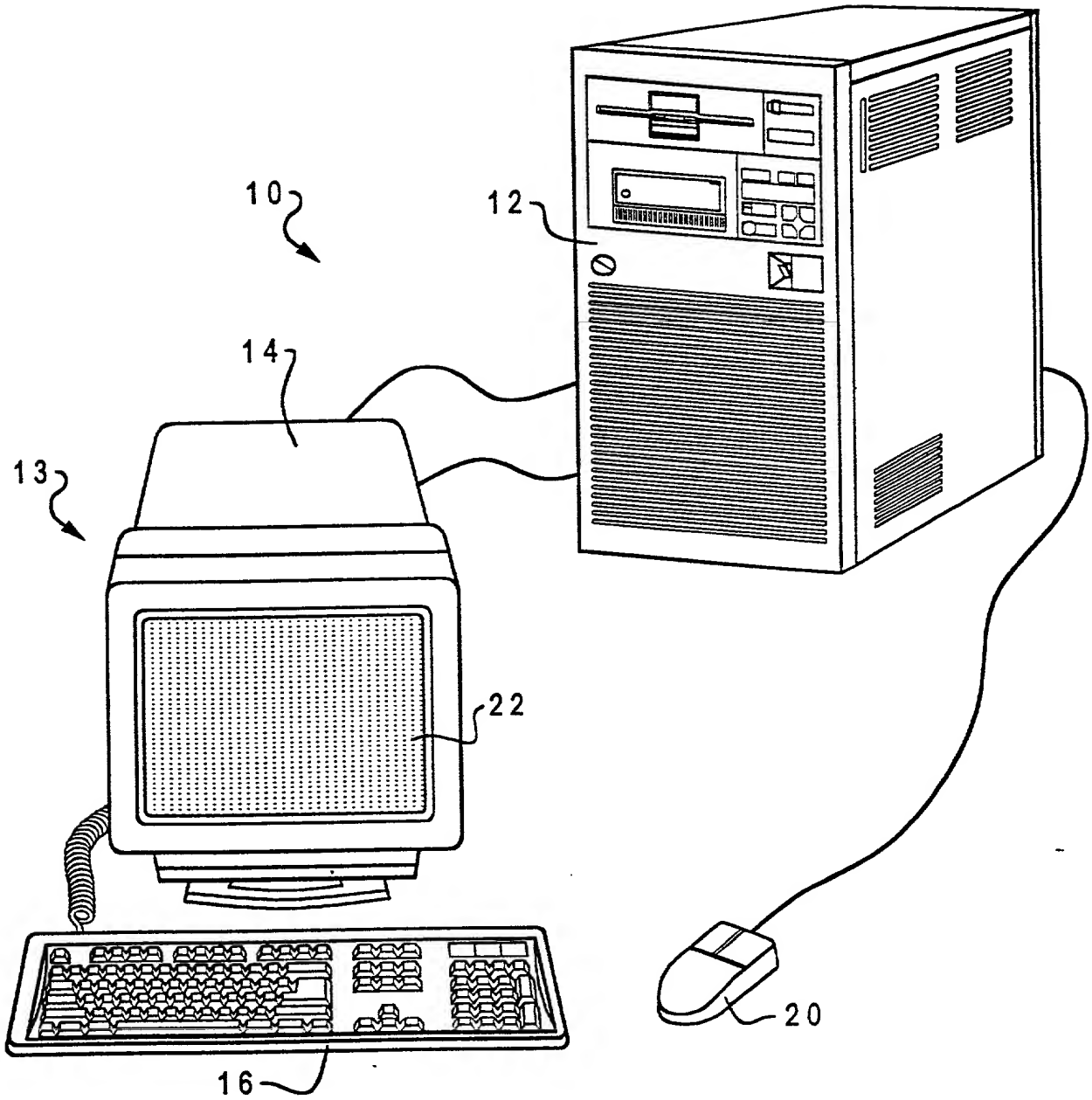

Fig. 1

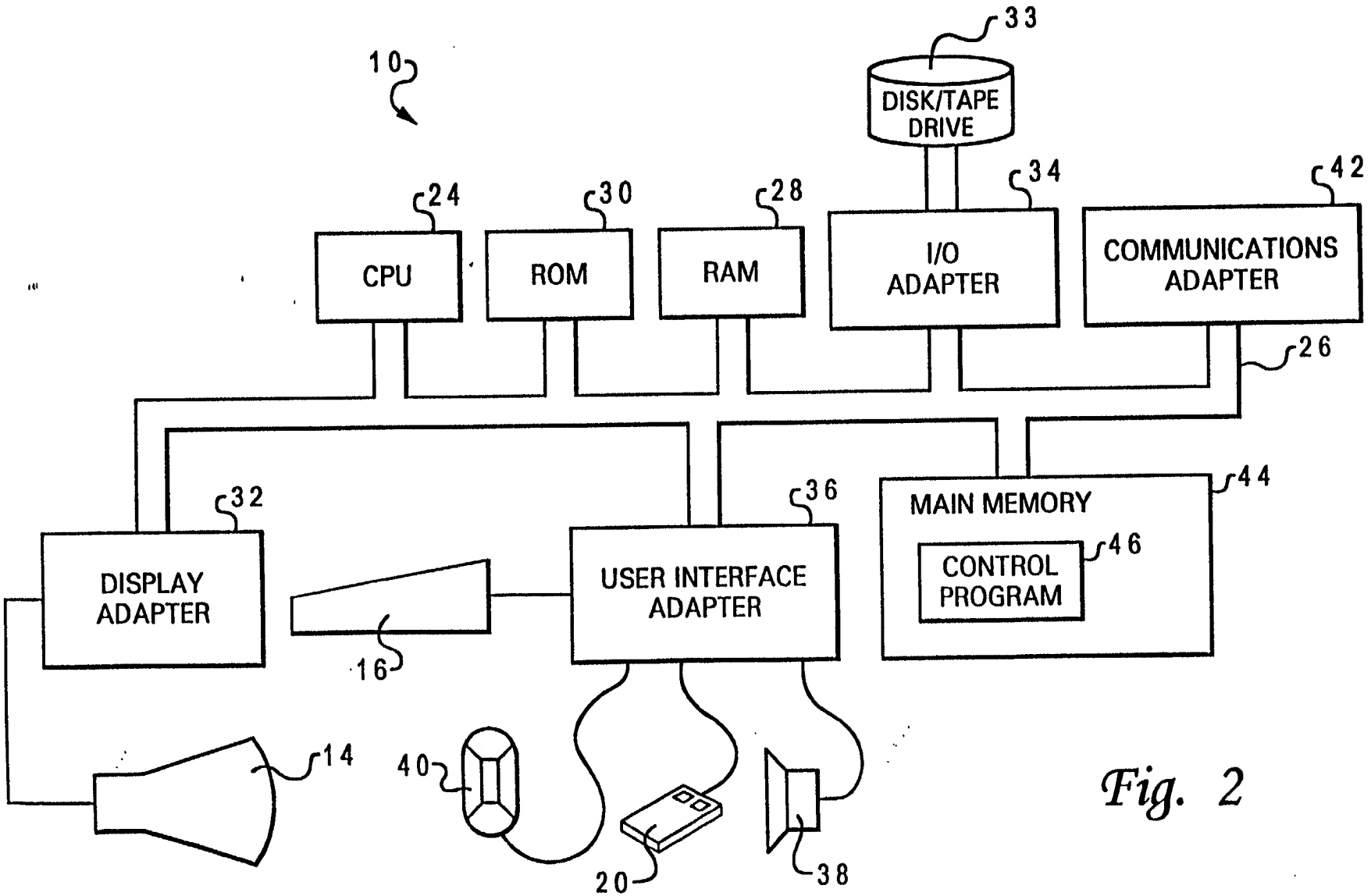
10
33
DISK/TAPE DRIVE
34
42
24
30
28
CPU
ROM
RAM
I/O ADAPTER
COMMUNICATIONS ADAPTER
26
32
36
44
MAIN MEMORY
46
CONTROL PROGRAM
DISPLAY ADAPTER
USER INTERFACE ADAPTER
16
14
40
20
38

Fig. 2

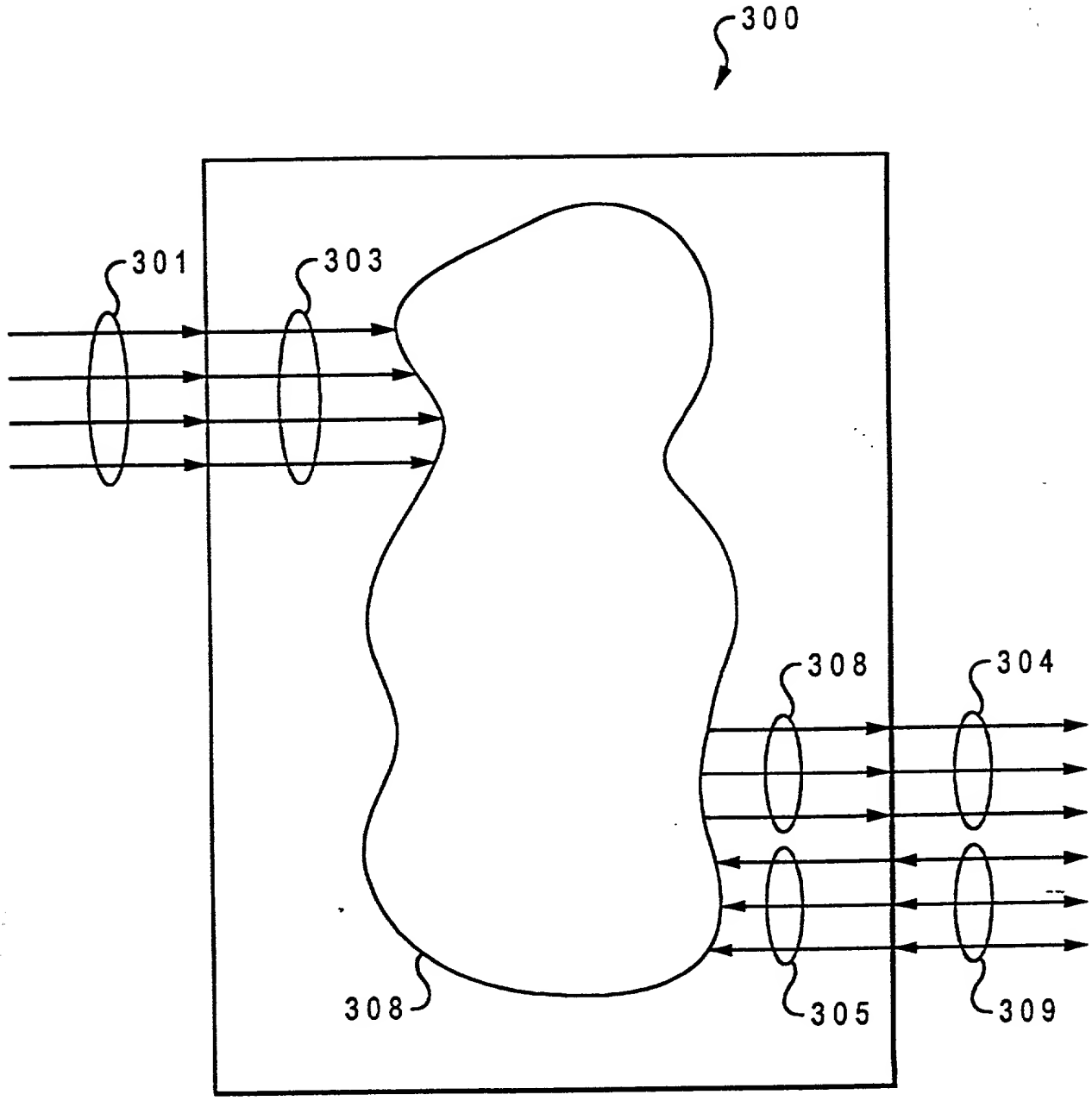

Fig. 3A

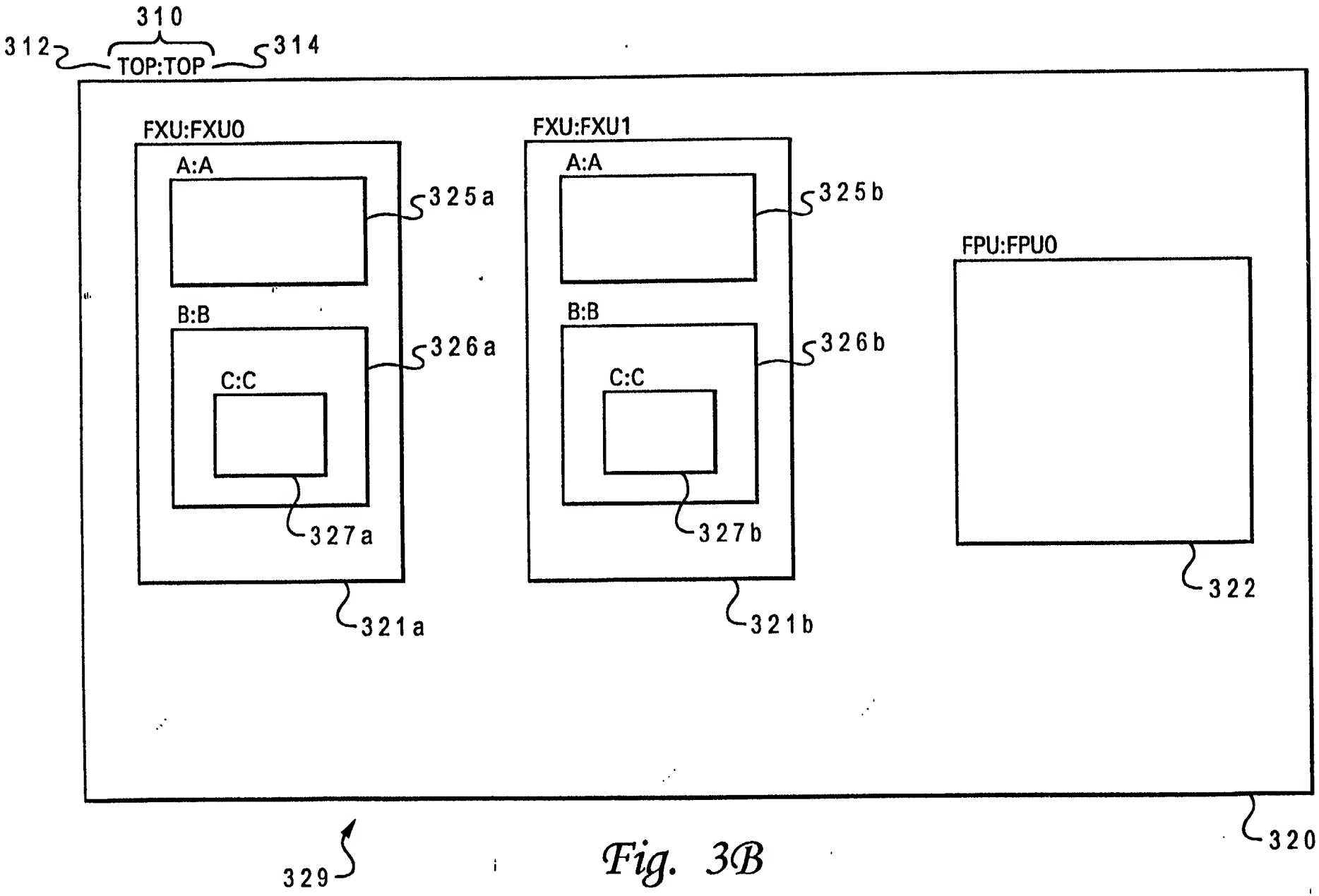
310
312
TOP:TOP
314
FXU:FXU0
A:A
325a
B:B
326a
C:C
327a
321a
FXU:FXU1
A:A
325b
B:B
326b
C:C
327b
321b
FPU:FPU0
322
320
329

Fig. 3B

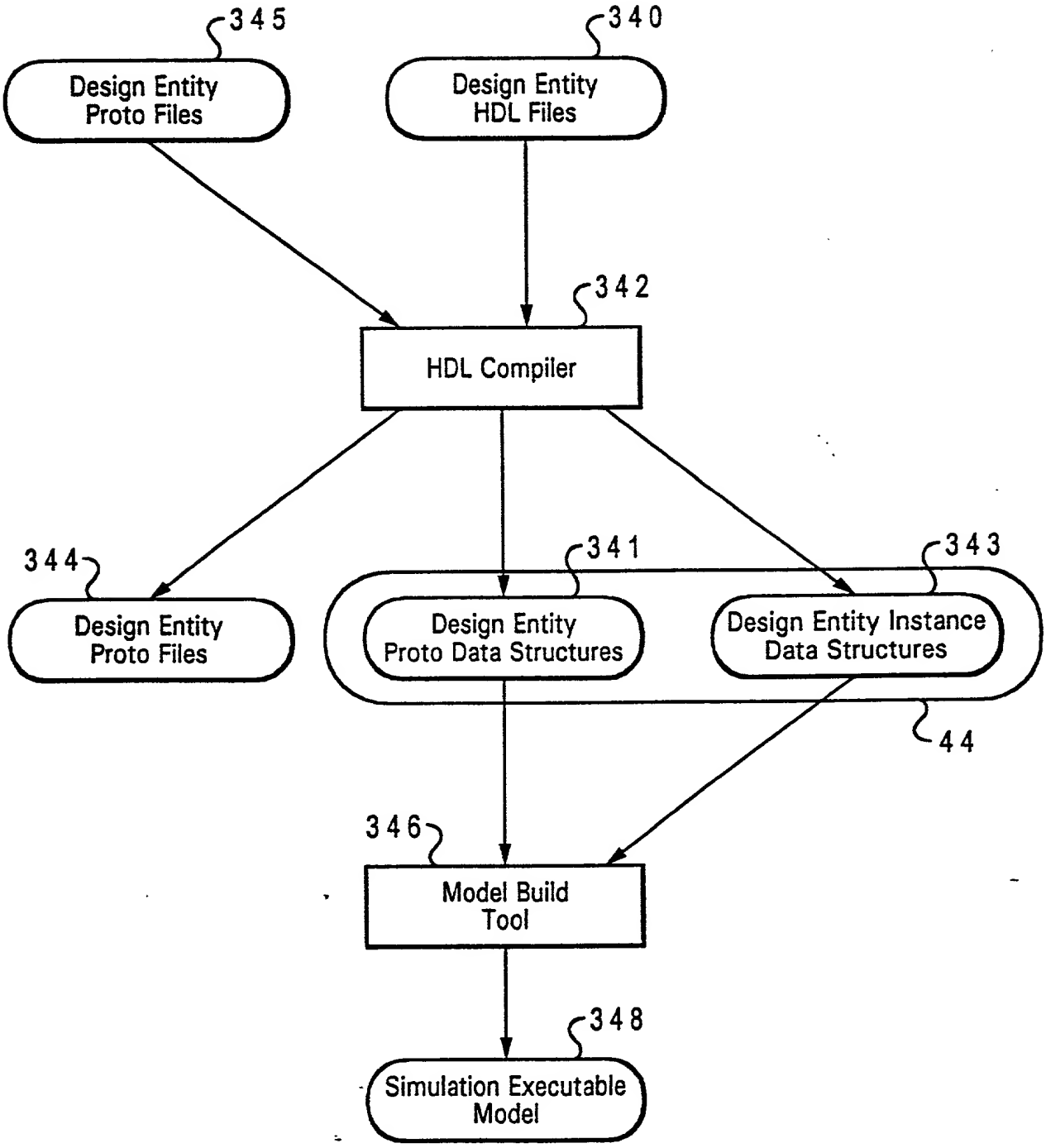
345
Design Entity Proto Files
340
Design Entity HDL Files
342
HDL Compiler
344
Design Entity Proto Files
341
Design Entity Proto Data Structures
343
Design Entity Instance Data Structures
44
346
Model Build Tool
348
Simulation Executable Model

Fig. 3C

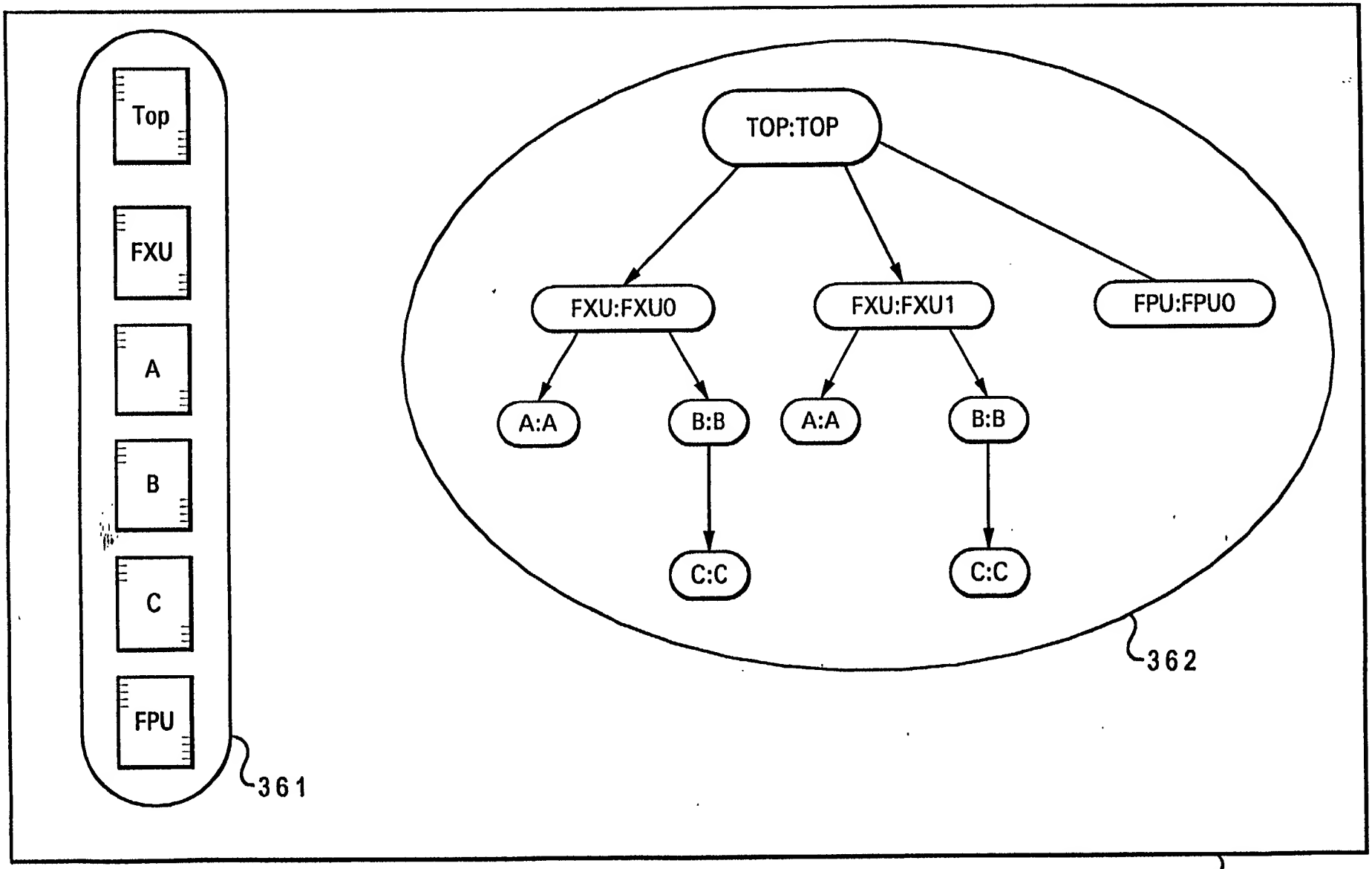
Top
FXU
A
B
C
FPU
361
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
B:B
A:A
B:B
C:C
C:C
362
44

Fig. 3D

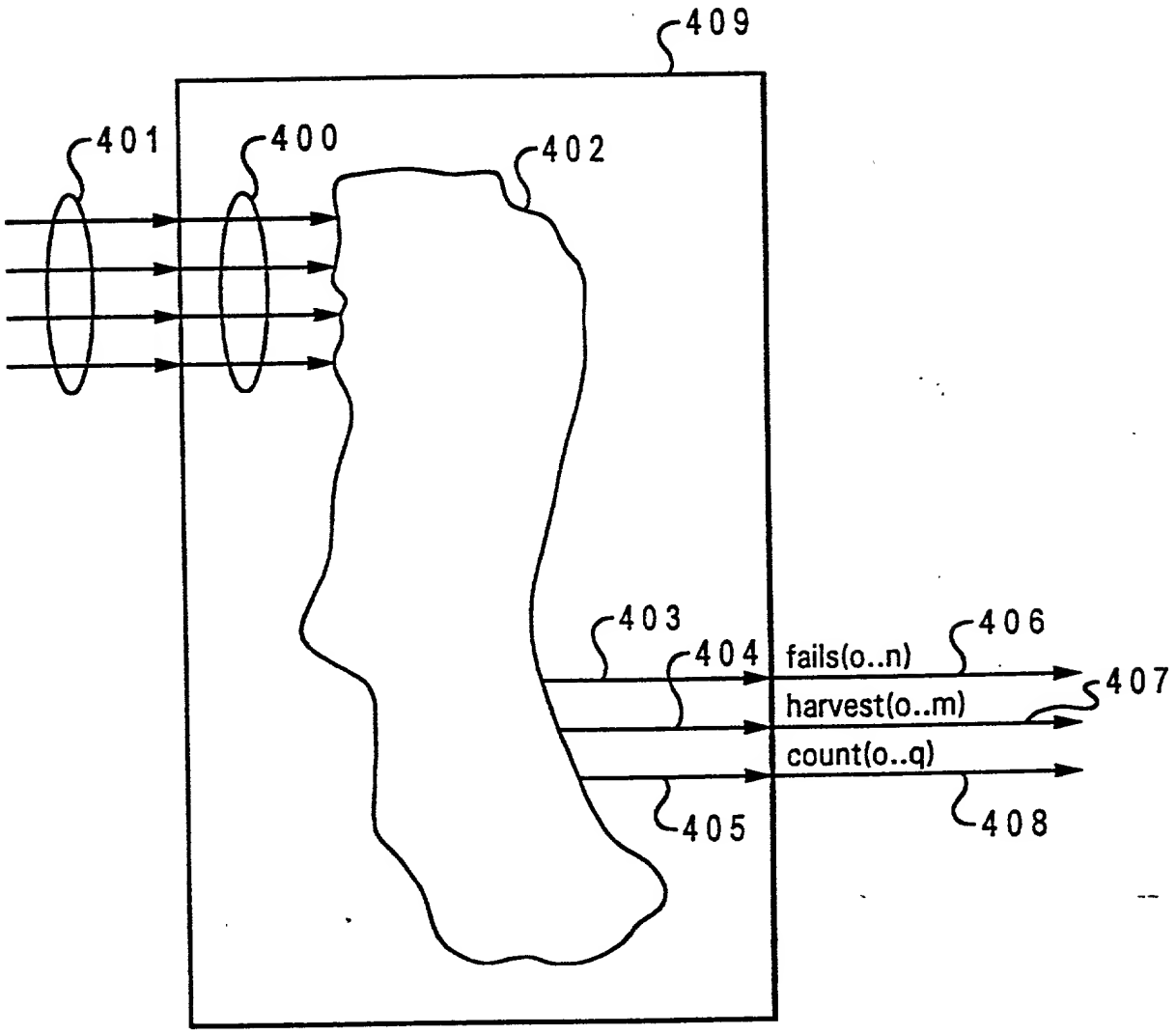
409
401
400
402
403
404
406
fails(o..n)
407
harvest(o..m)
count(o..q)
405
408

Fig. 4A

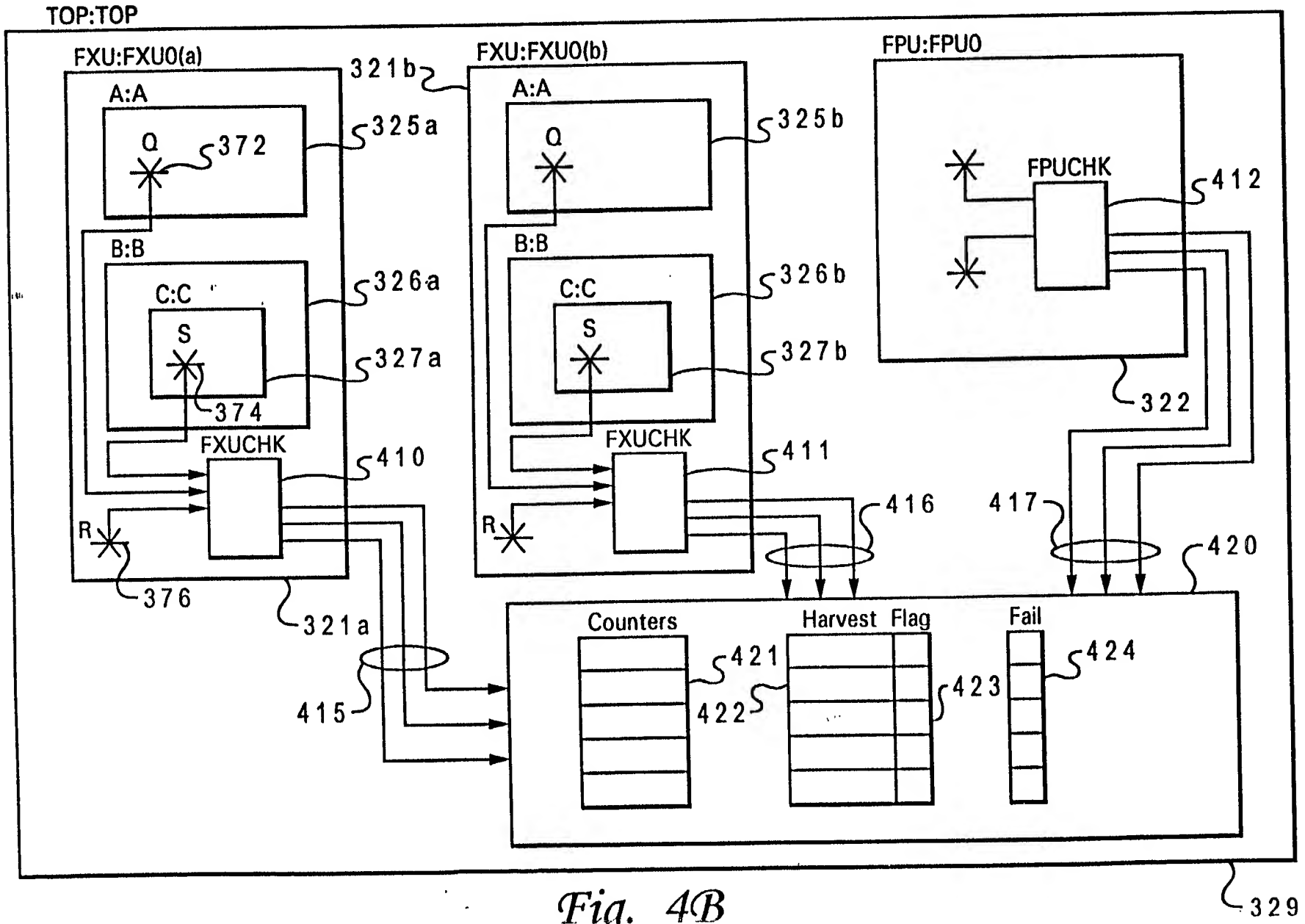

Fig. 4B

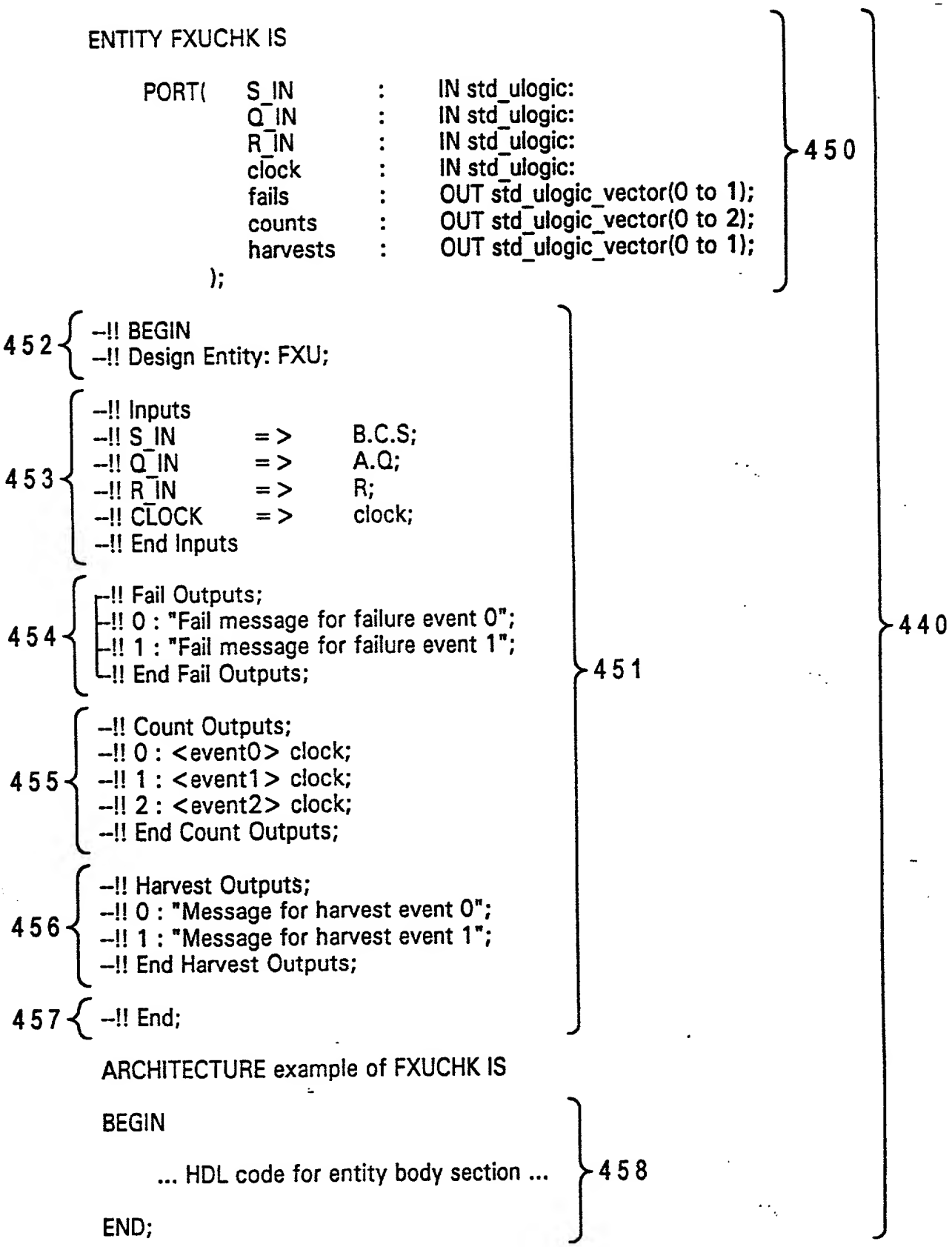
ENTITY FXUCHK IS
PORT(S_IN : IN std_ulogic:
Q_IN : IN std_ulogic:
R_IN : IN std_ulogic:
clock : IN std_ulogic:
fails : OUT std_ulogic_vector(0 to 1);
counts : OUT std_ulogic_vector(0 to 2);
harvests : OUT std_ulogic_vector(0 to 1);
);
450
452
--!! BEGIN
--!! Design Entity: FXU;
453
--!! Inputs
--!! S_IN => B.C.S;
--!! Q_IN => A.Q;
--!! R_IN => R;
--!! CLOCK => clock;
--!! End Inputs
454
--!! Fail Outputs;
--!! 0 : "Fail message for failure event 0";
--!! 1 : "Fail message for failure event 1";
--!! End Fail Outputs;
451
455
--!! Count Outputs;
--!! 0 : <event0> clock;
--!! 1 : <event1> clock;
--!! 2 : <event2> clock;
--!! End Count Outputs;
456
--!! Harvest Outputs;
--!! 0 : "Message for harvest event 0";
--!! 1 : "Message for harvest event 1";
--!! End Harvest Outputs;
457
--!! End;
ARCHITECTURE example of FXUCHK IS
BEGIN
... HDL code for entity body section ...
458
END;
440

Fig. 4C

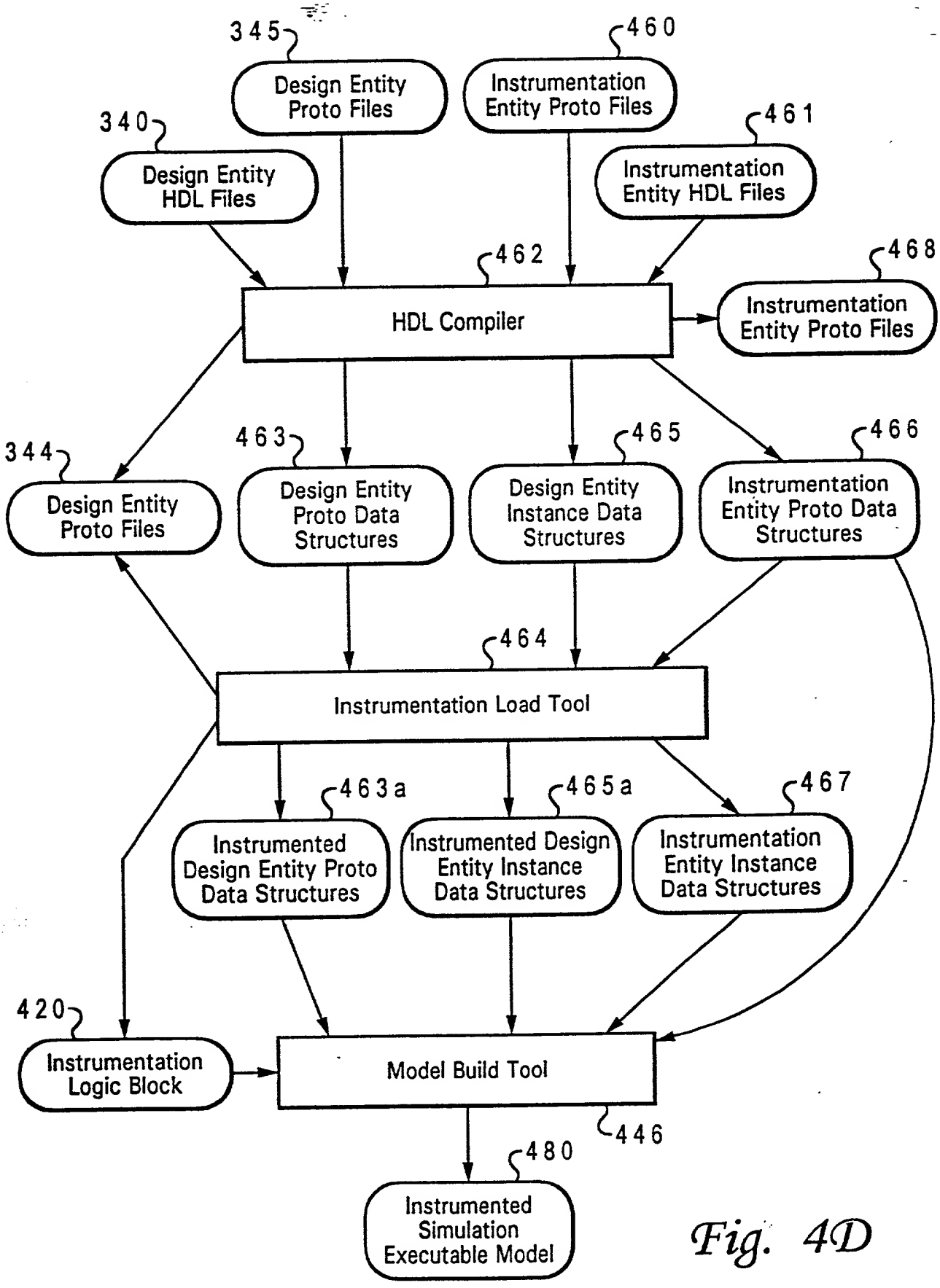
345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

Fig. 4D

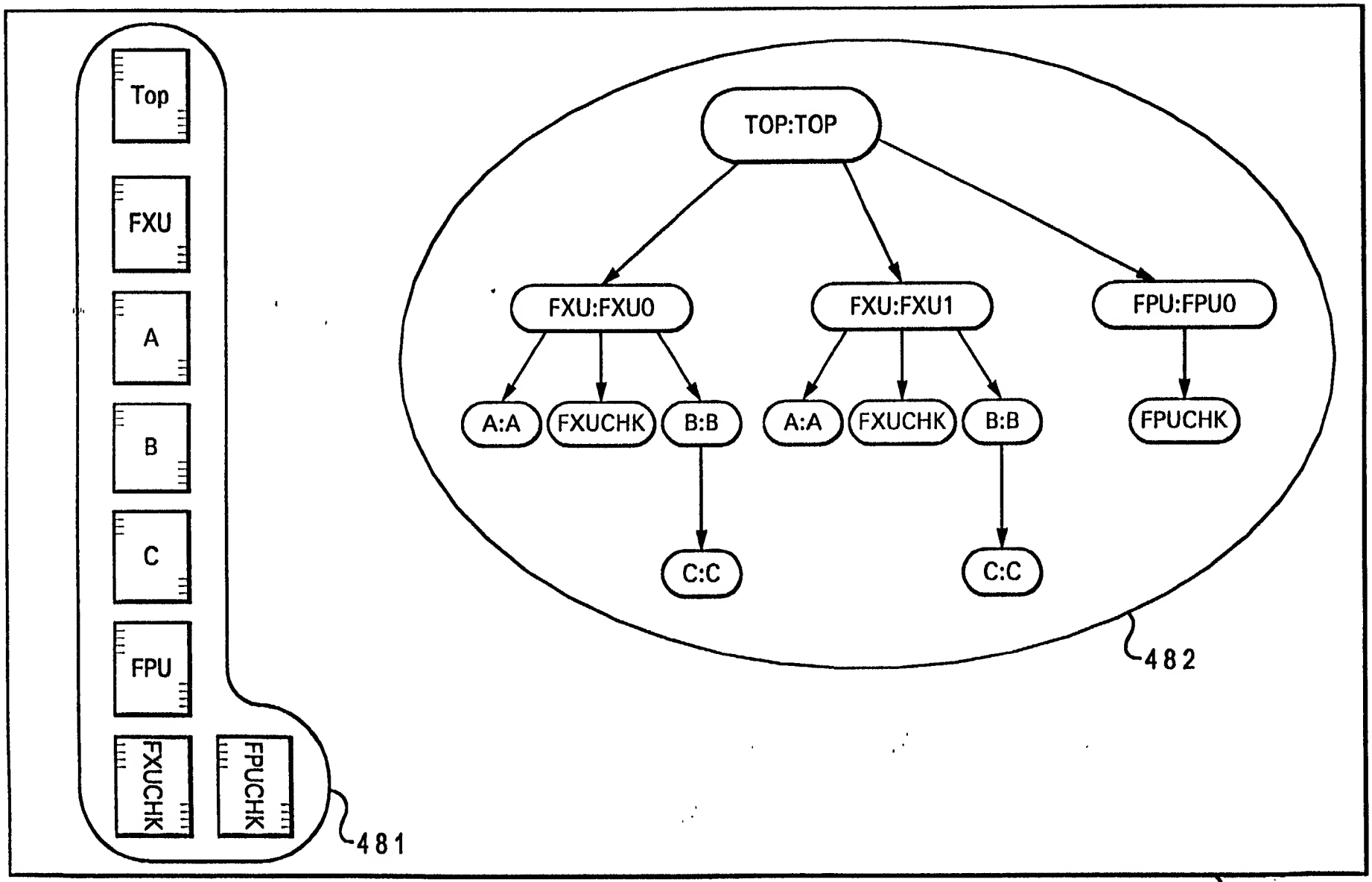
Top
FXU
A
B
C
FPU
FXUCHK
FPUCHK
481
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
A:A
FXUCHK
B:B
FPUCHK
C:C
C:C
482
44

Fig. 4E

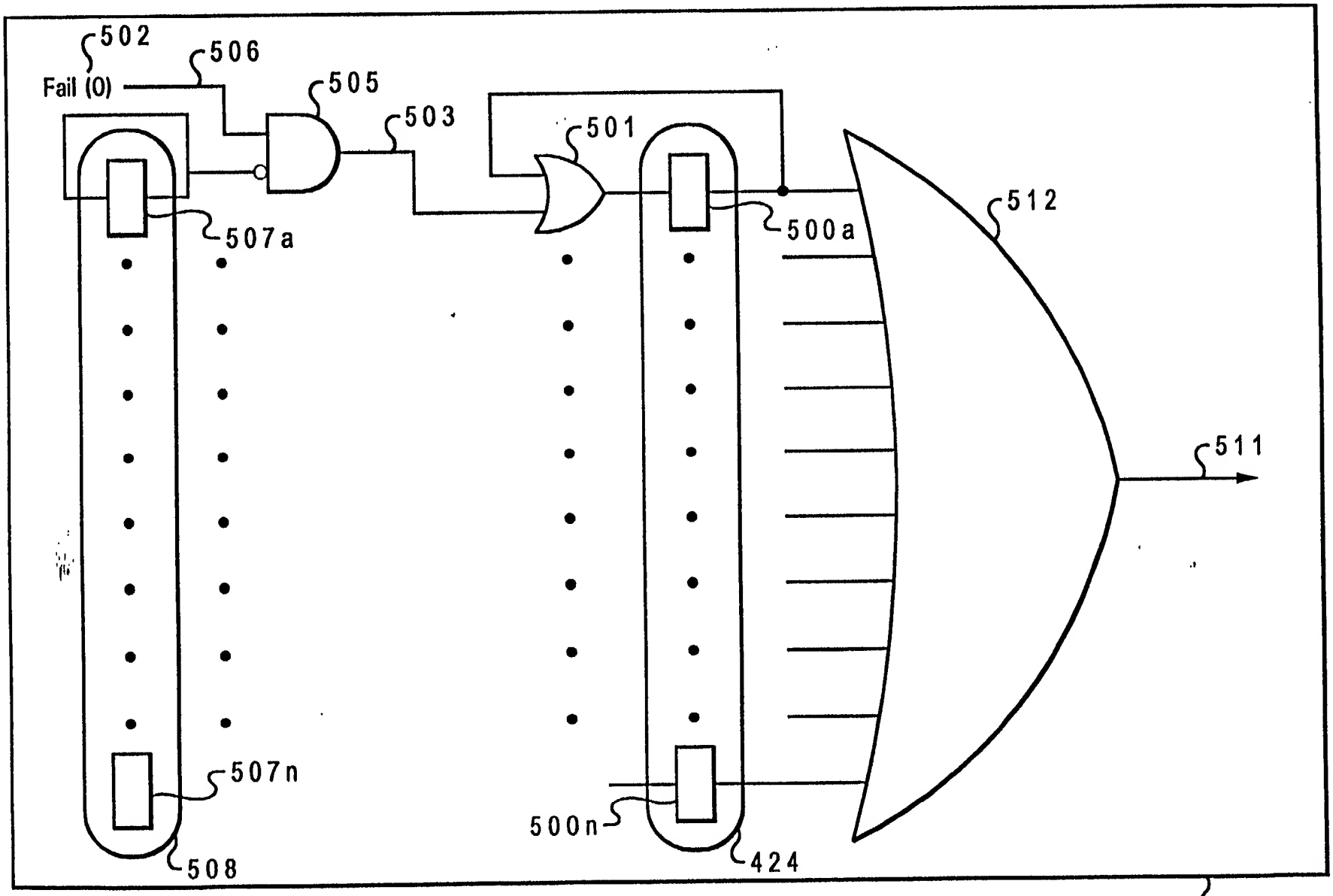
502
506
Fail (0)
505
503
501
507a
500a
512
511
507n
500n
508
424
420

Fig. 5A

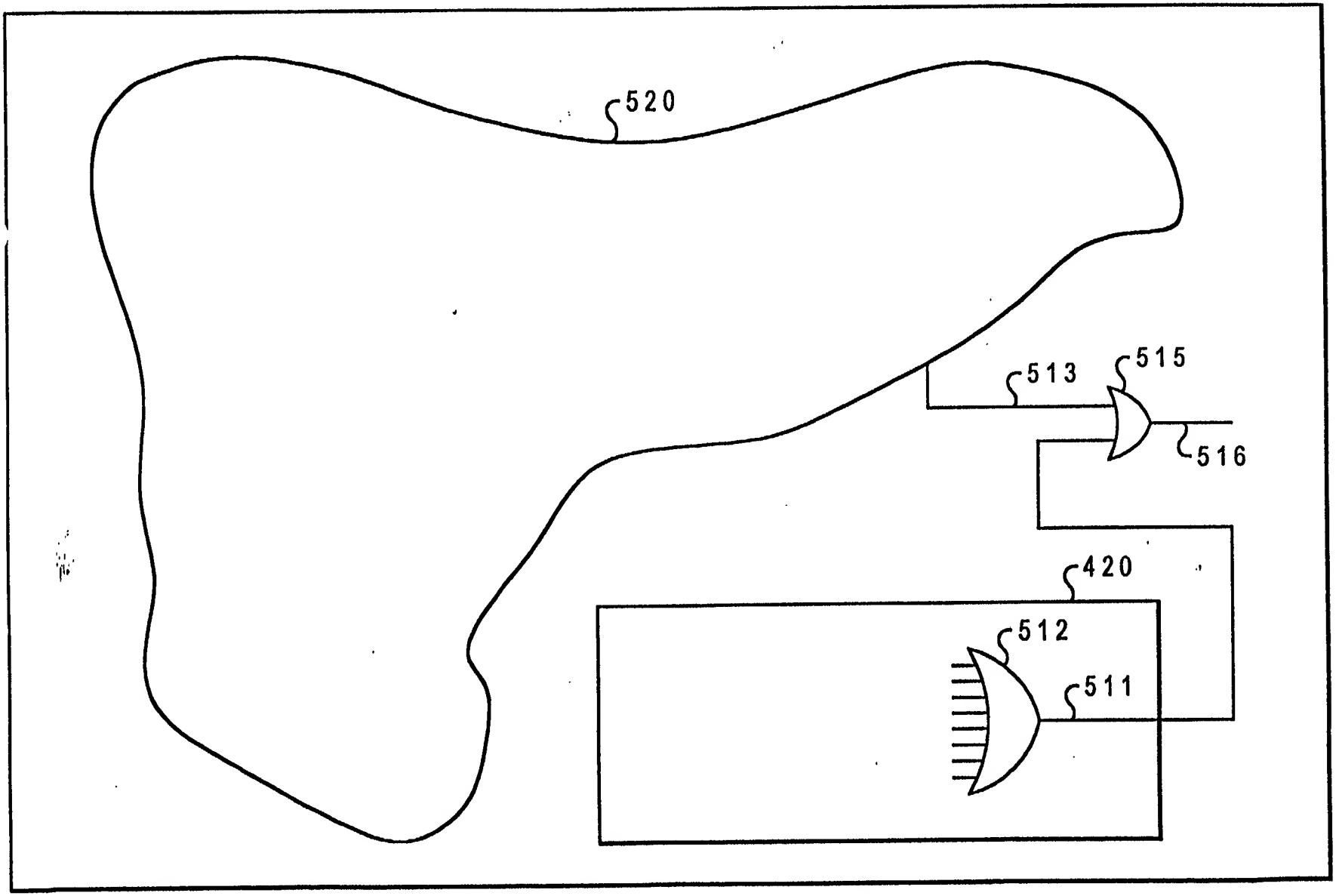

Fig. 5B

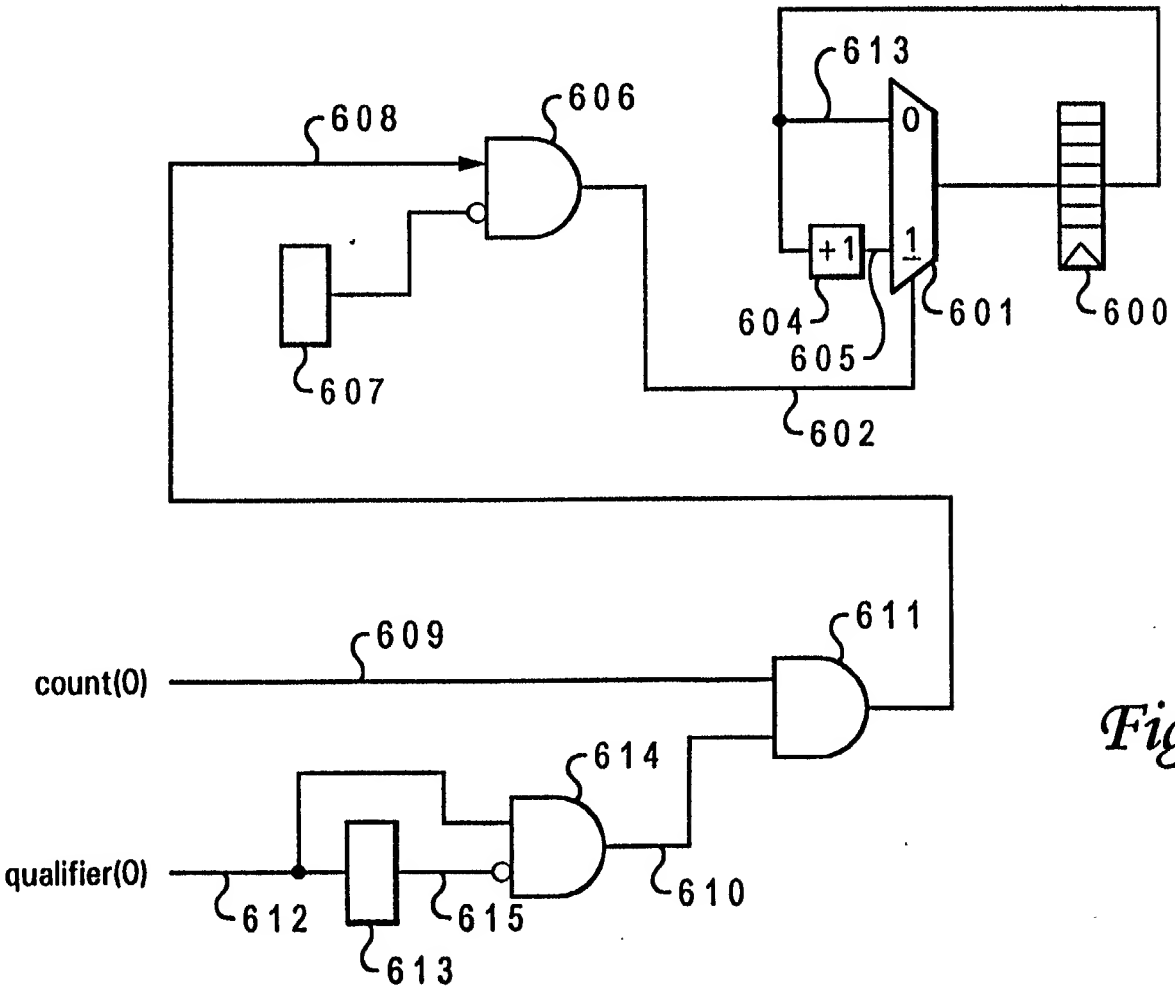
613
608
606
0
+1
1
604
605
601
600
607
602
611
609
count(0)
614
qualifier(0)
612
615
610
613

Fig. 6A

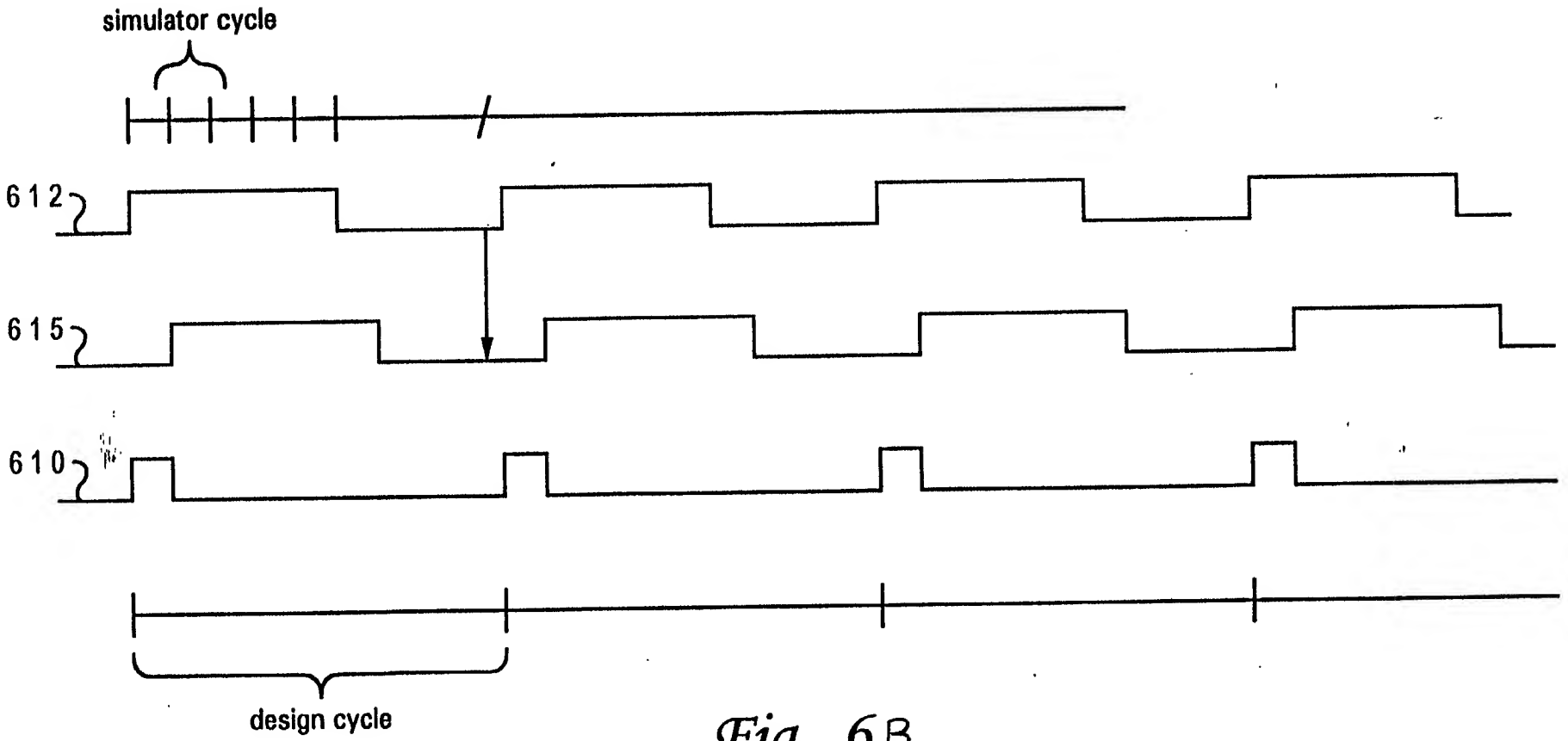

Fig. 6B

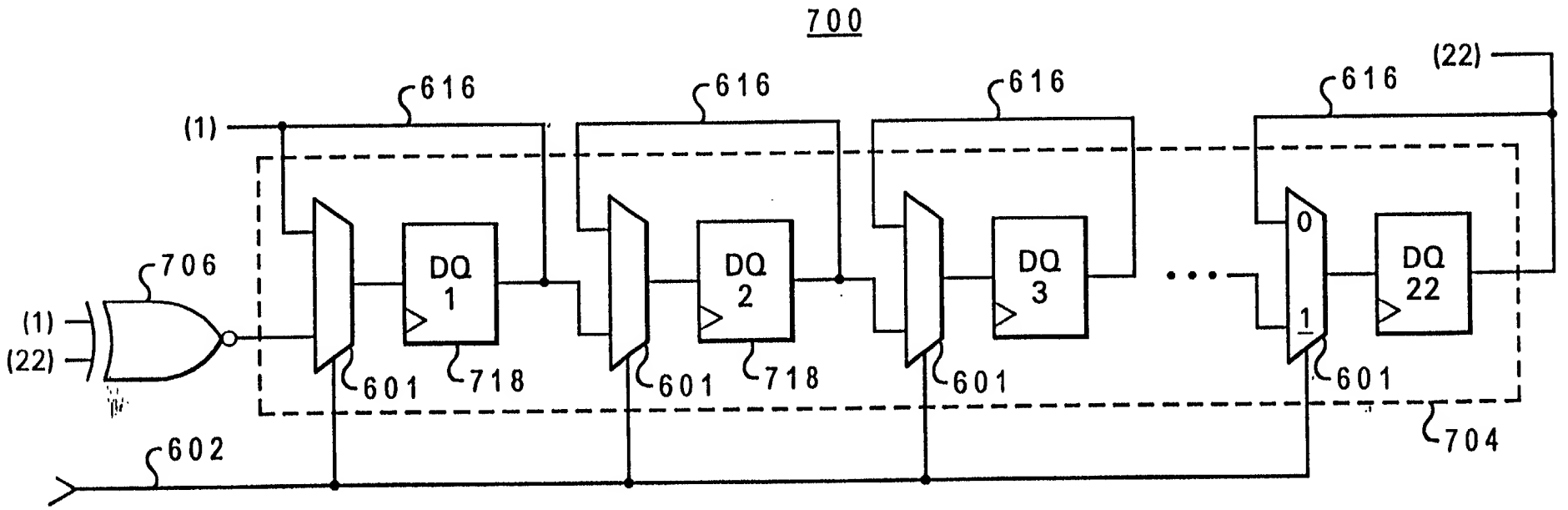

Fig. 7

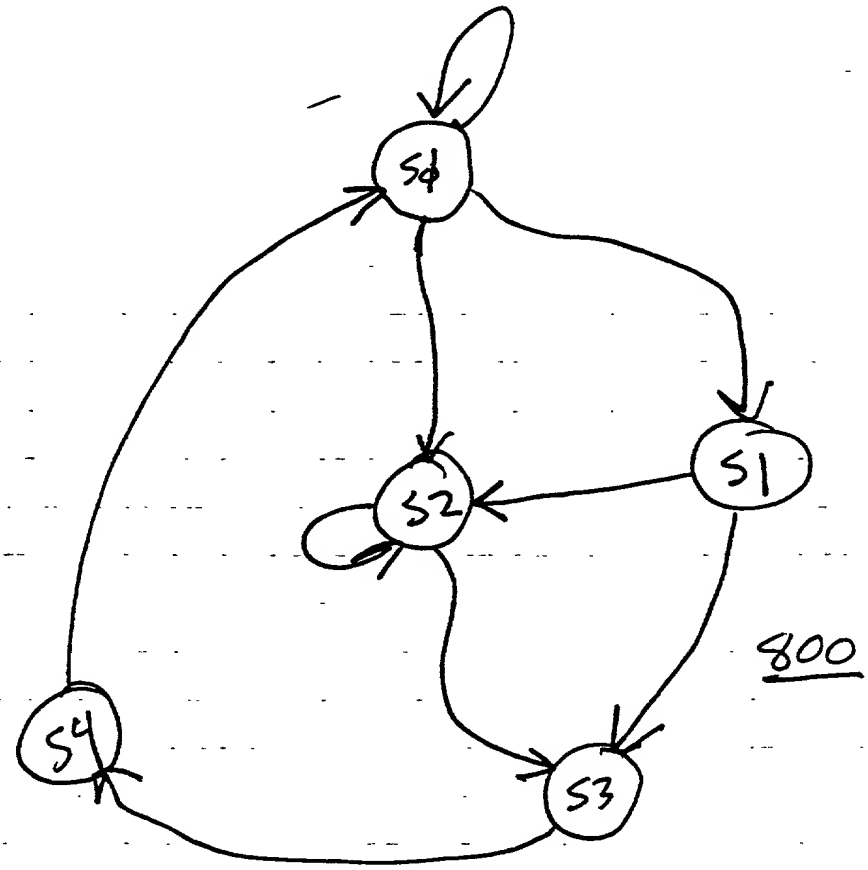

FIG. 8

(Prior Art)

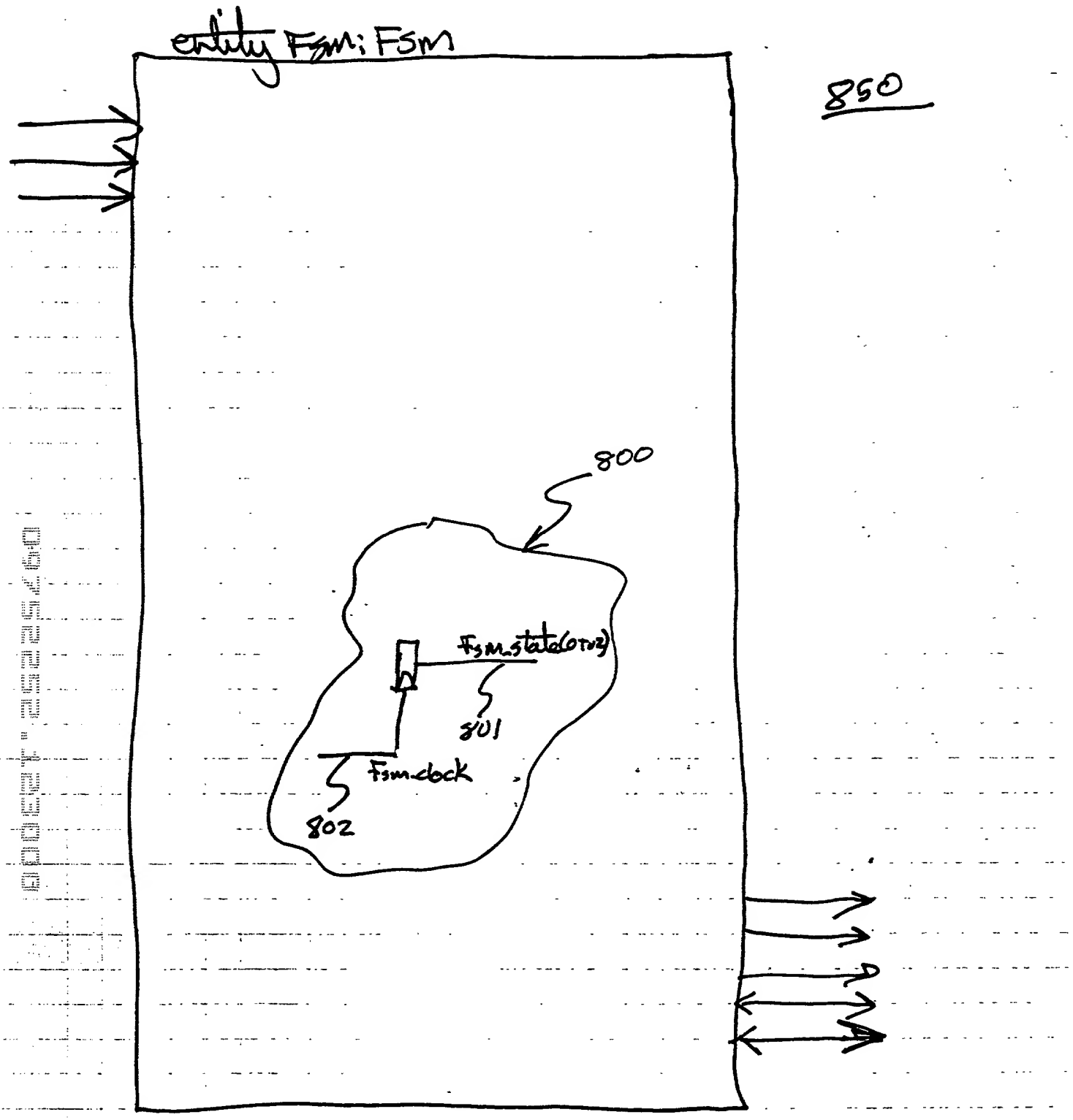

FIG. 8A
(Prior Art)

```
entity Fsm IS
  PORT(
        .... ports for entity Fsm .....
      );
  ARCHITECTURE Fsm of Fsm IS
  BEGIN
        .... HDL code For Fsm and rest of the entity...
        fsm_state(0 to 2) <= ... signal 801 ....

853 {   --!! Embedded Fsm : exampleFsm;
859 {   --!! clock        : (fsm_clock);
854 {   --!! state_vector : (fsm_state(0 to 2));
855 {   --!! states       : (s0, s1, s2, s3, s4);
856 {   --!! state_encoding : ('000', '001', '010', '011', '100');
857 {   --!! arcs           : (s0=>s0, s0=>s1, s0=>s2,
        --!!                  s1=>s2, s1=>s3, s2=>s2,
        --!!                  s2=>s3, s3=>s4, s4=>s0);
858 {   --!! end Fsm;

  END;
```

852

86

FIG. 8B

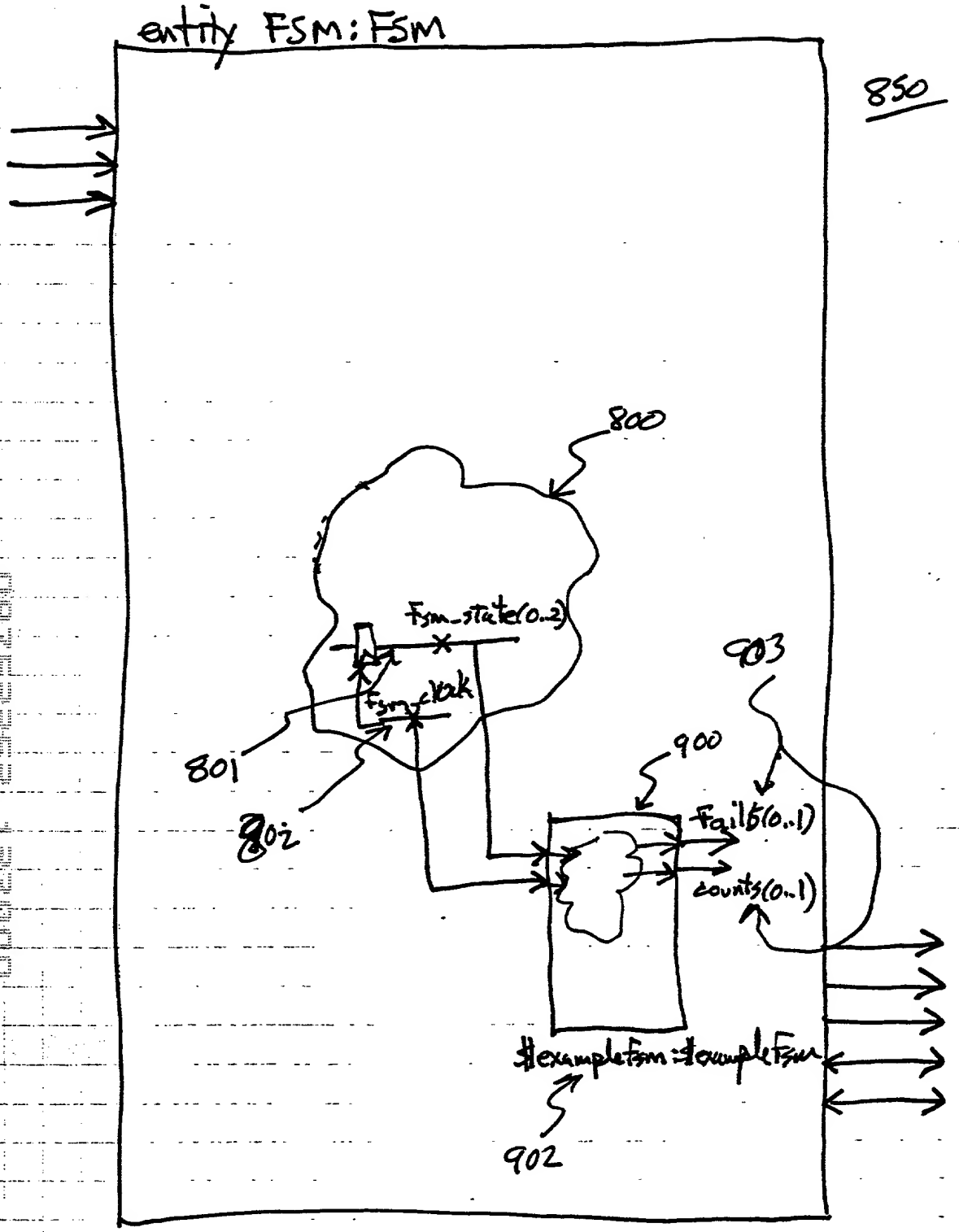
entity FSM:FSM
850
800
fsm_state(0..2)
fsm_clock
801
802
903
900
Failed(0..1)
counts(0..1)
$exampleFsm:exampleFsm
902

FIG. 9

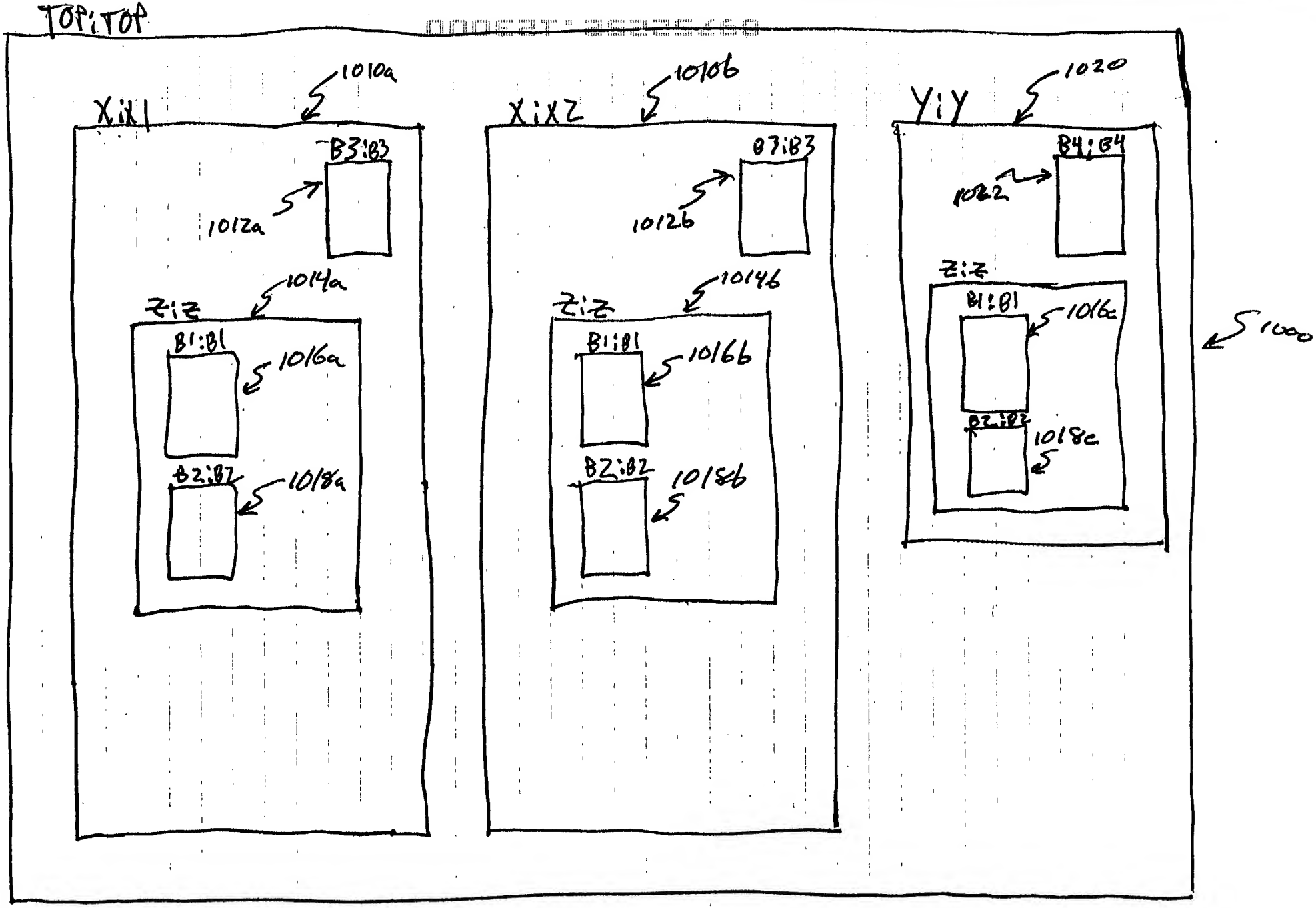
TOP:TOP
X:X1
1010a
B3:B3
1012a
Z:Z
1014a
B1:B1
1016a
B2:B2
1018a
X:X2
10106
B3:B3
10126
Z:Z
10146
B1:B1
10166
B2:B2
10186
Y:Y
1020
B4:B4
1022
Z:Z
B1:B1
1016c
B2:B2
1018c
1000

FIG. 10A

1030 1032 1034 1036

<instantiation identifier> . <instrumentation entity name> . <design entity name> . <event name>

FIG 10B

1030	1032	1034	1036	
X1	B3	X	COUNT1	1040
X1.Z	B1	Z	COUNT7	1041
X1.Z	B2	Z	COUNT1	1042
X2	B3	X	COUNT1	1043
X2.Z	B1	Z	COUNT4	1044
X2.Z	B2	Z	COUNT1	1045
Y	B4	Y	COUNT1	1046
Y.Z	B1	Z	COUNT1	1047
Y.Z	B2	Z	COUNT1	1048

FIG 10C

1030 1034 1036

<instantiation identifier> . <design entity name> . <event name>

FIG. 10D

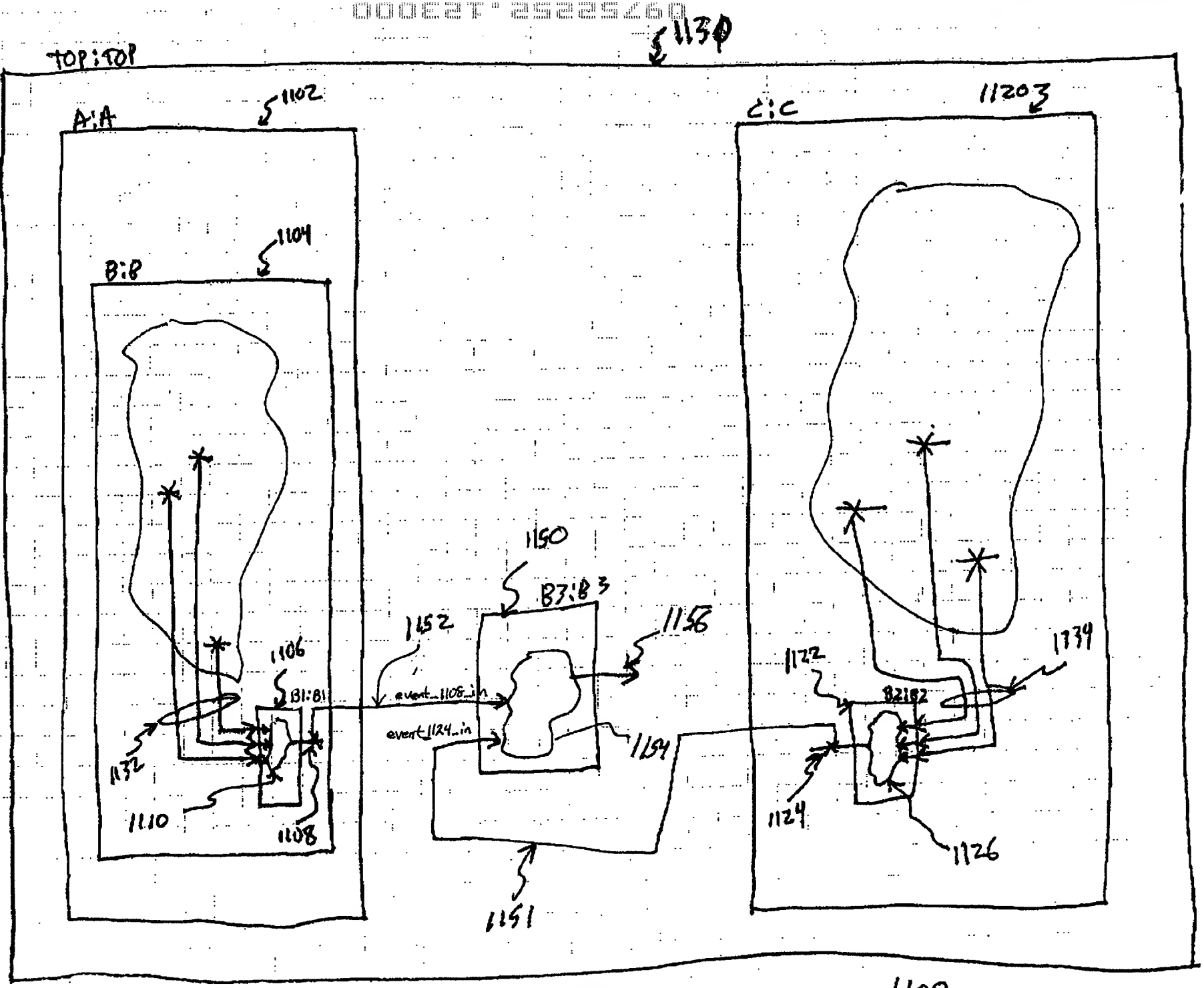

FIG. 11A

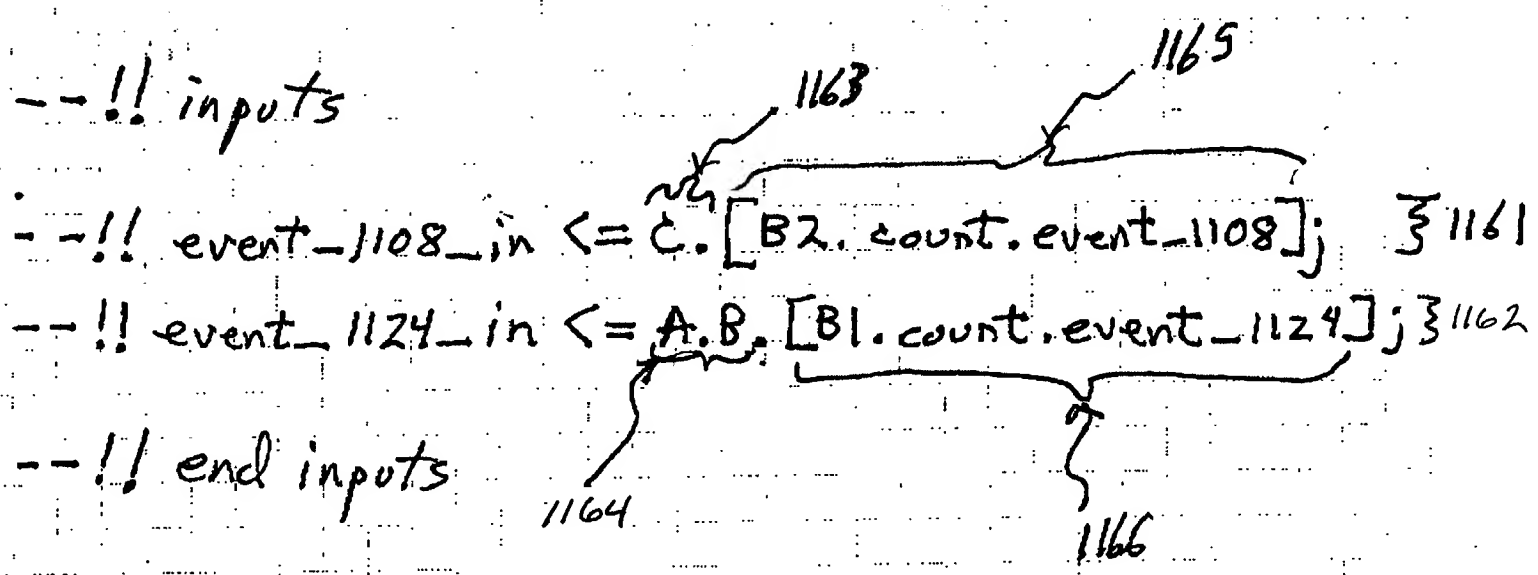

FIG. 11B

```
--!! inputs
--!! event_1108_in <= C.[count.event_1108]; } 1171
--!! event_1124_in <=  B.[count.event_1124]; } 1172
--!! end inputs
```

FIG. 11C

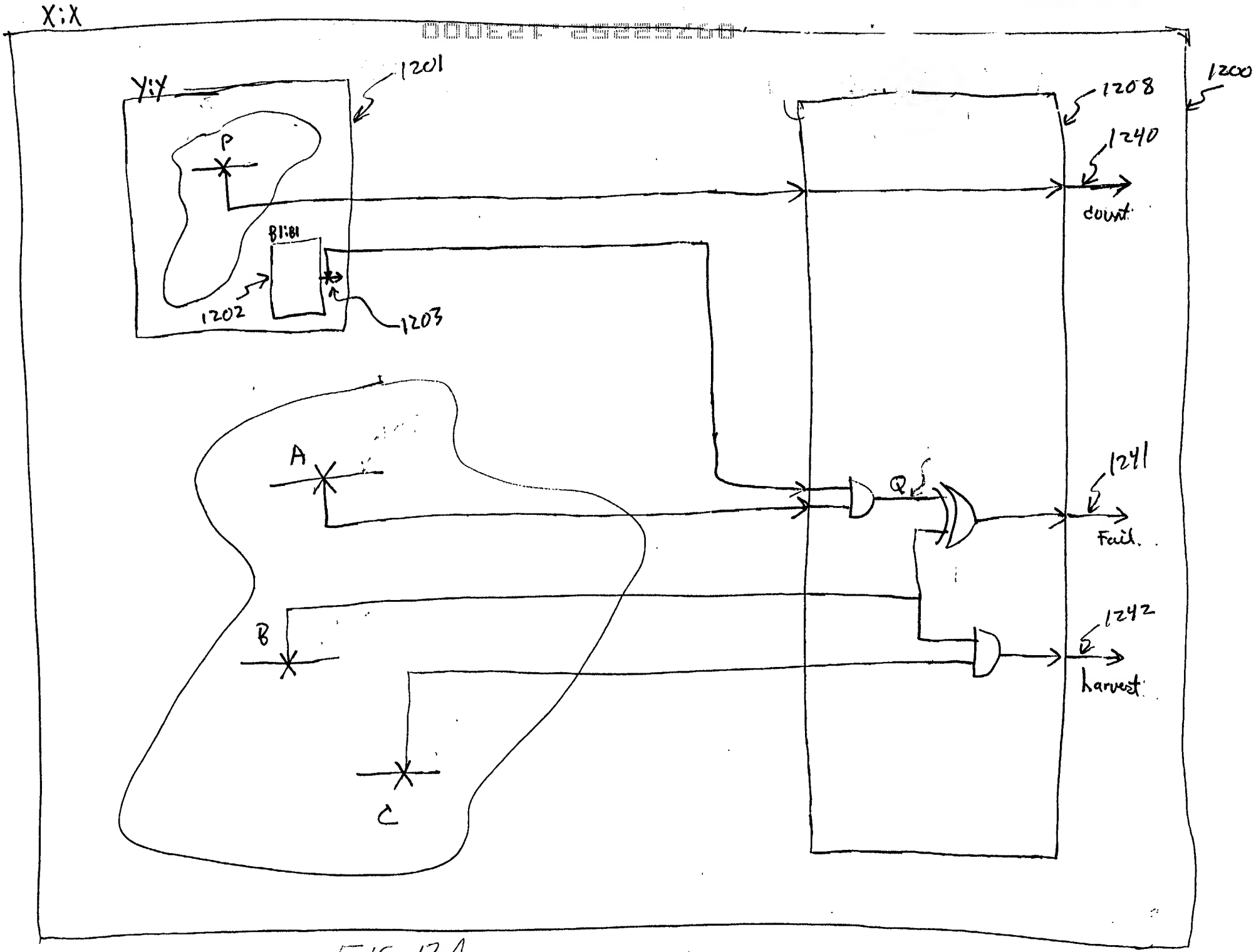
X:X
Y:Y
1201
1200
1208
1240
count
P
B1:B1
1202
1203
A
B
C
Q
1241
Fail
1242
harvest

FIG. 12A

```
Entity X IS
     PORT ( :
            :
            :
          );

ARCHITECTURE example OF X IS
BEGIN
     :
     :
   ...HDL CODE For X.....
     :
     :

   Y:Y
   PORT MAP( :
             :
           );                                              } 1221

   A <= ....
   B <= ....                                               } 1222
   C <= ....

--!! [count, countname0, clock] <= Y.P;                    } 1230
--!! Q <= Y.[B1.count .count1] AND A;                      } 1232
--!! [fail, Failname0, "fail msg"] <= Q XOR B;             } 1234
--!! [harvest, harvestname0, "harvest msg"] <= B AND C;    } 1236   } 1223

END
```

FIG. 12B

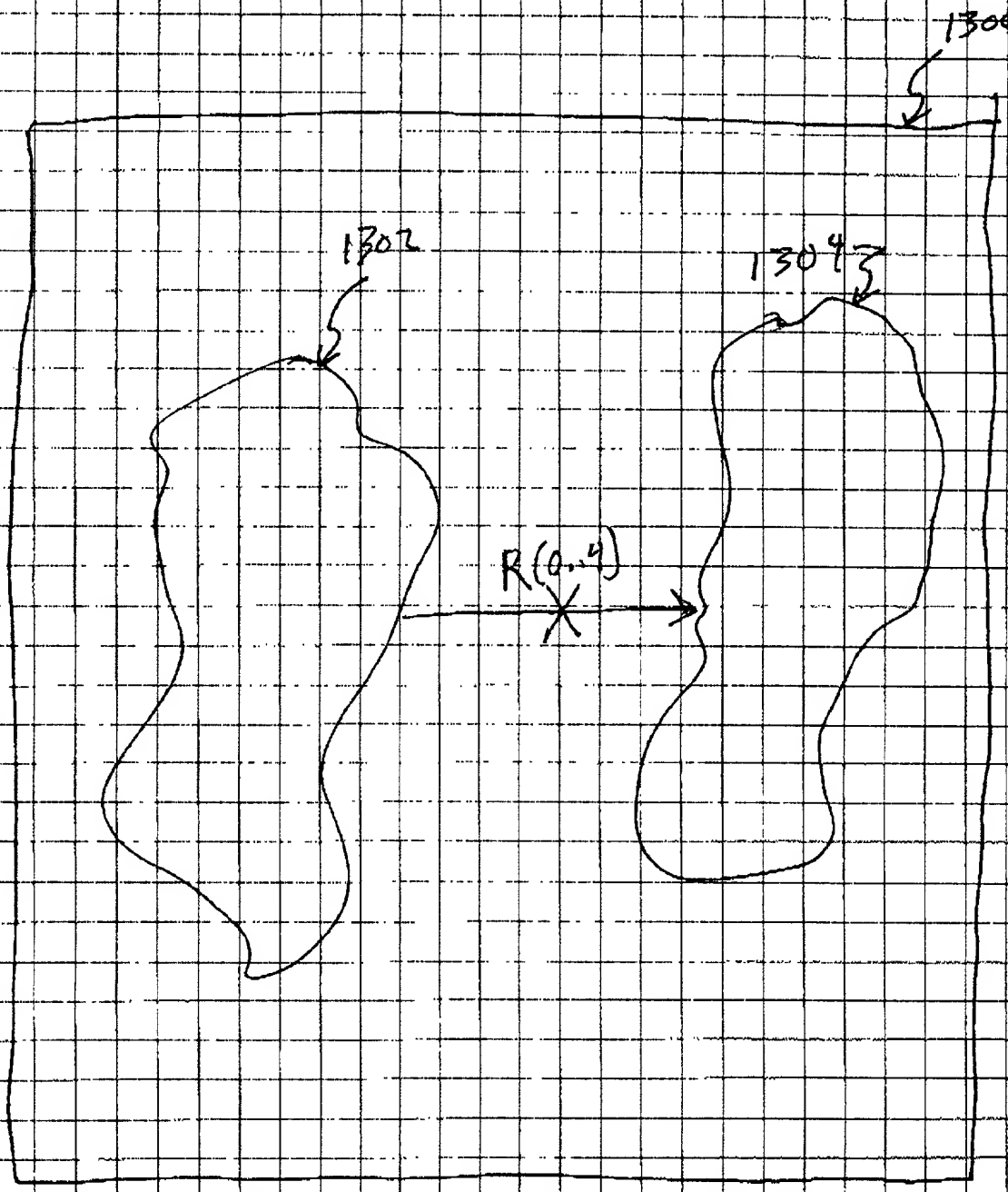

FIG. 13A

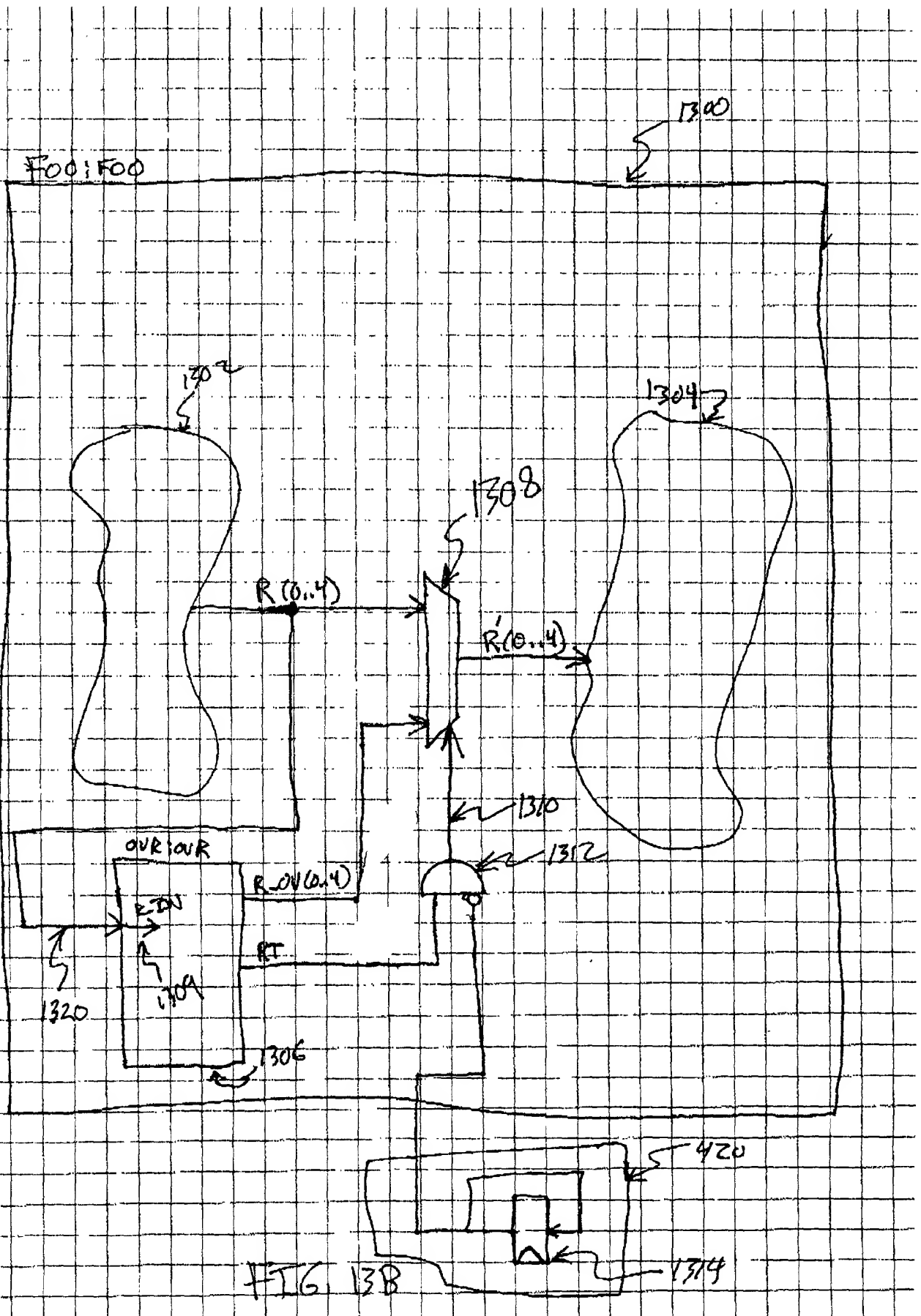

FIG. 13B

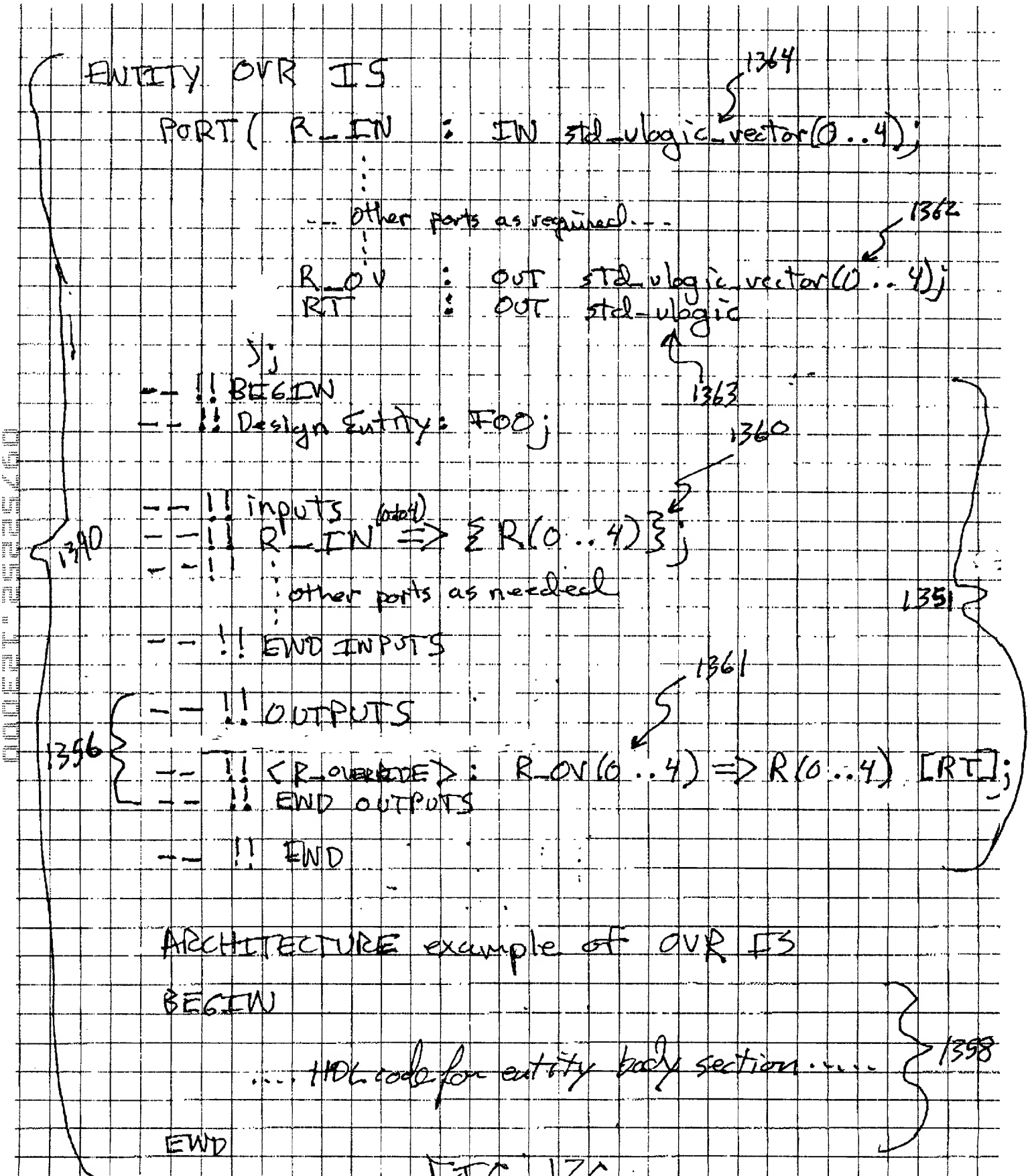
ENTITY OVR IS
PORT(R_IN : IN std_ulogic_vector(0..4);
-- other ports as required --
R_OV : OUT std_ulogic_vector(0..4);
RT : OUT std_ulogic
);
-- !! BEGIN
-- !! Design Entity: FOO;
-- !! inputs
-- !! R_IN (data) => {R(0..4)};
-- !! other ports as needed
-- !! END INPUTS
-- !! OUTPUTS
-- !! <R_OVERRIDE>: R_OV(0..4) => R(0..4) [RT];
-- !! END OUTPUTS
-- !! END
ARCHITECTURE example of OVR IS
BEGIN
.... HDL code for entity body section ....
END
1364
1362
1363
1360
1351
1340
1361
1356
1358

FIG. 13C

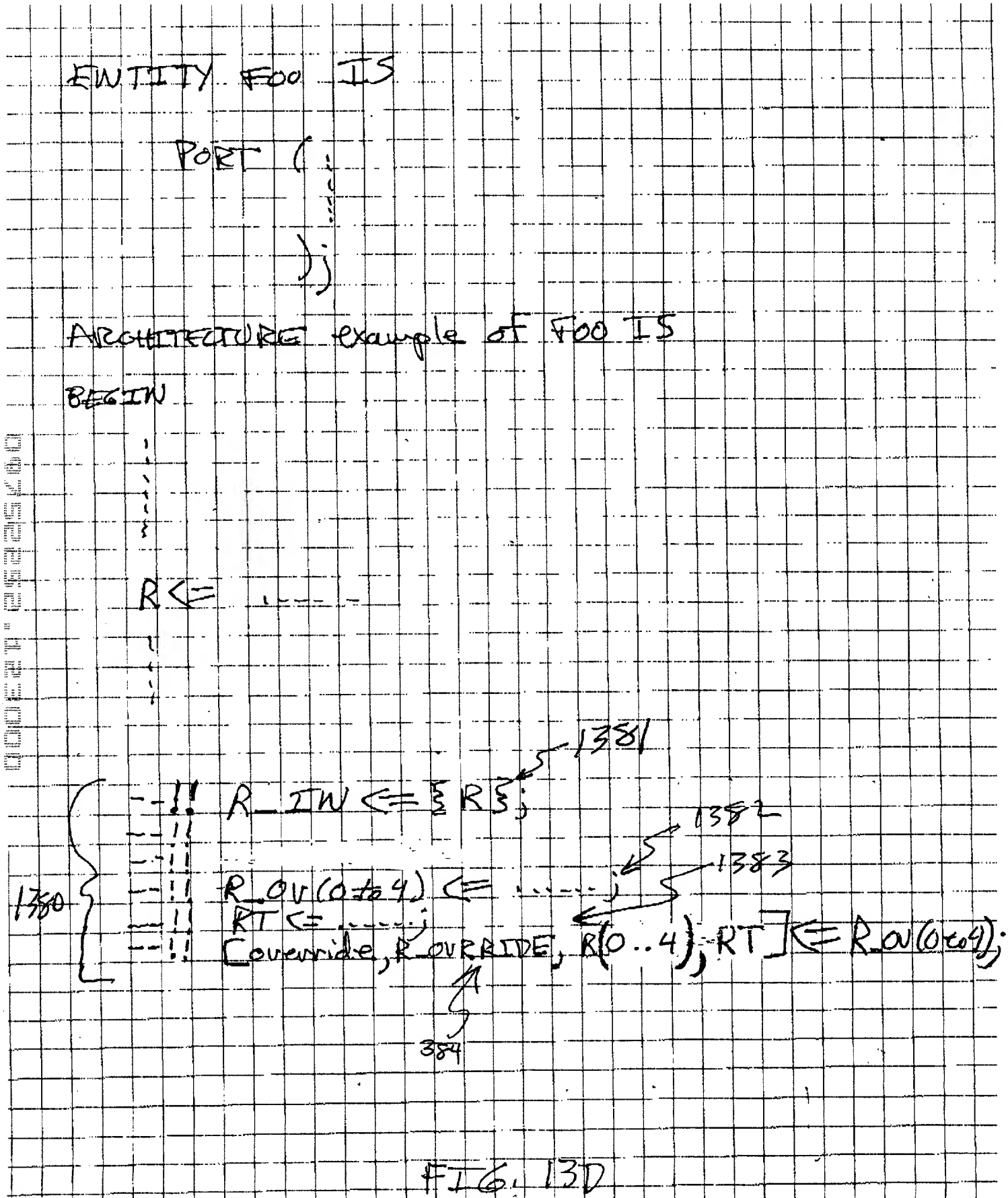
ENTITY FOO IS
PORT (
);
ARCHITECTURE example of FOO IS
BEGIN
R <= ....
R_IN <= R;
R_OV(0 to 4) <= ....;
RT <= ....;
[override, R_OVERRIDE, R(0..4), RT] <= R_OV(0 to 4);
1380
1381
1382
1383
384

FIG. 13D

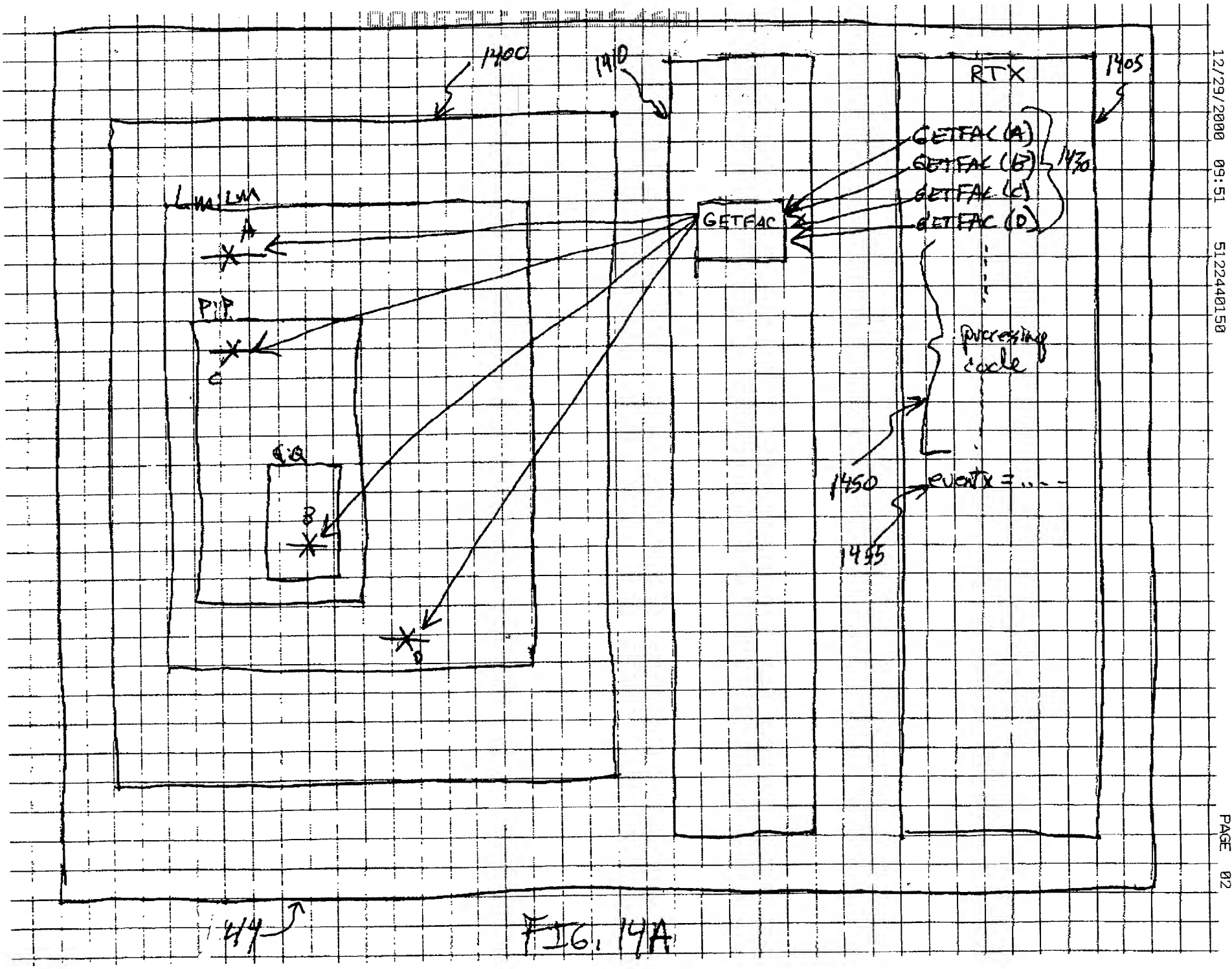

FIG. 14A

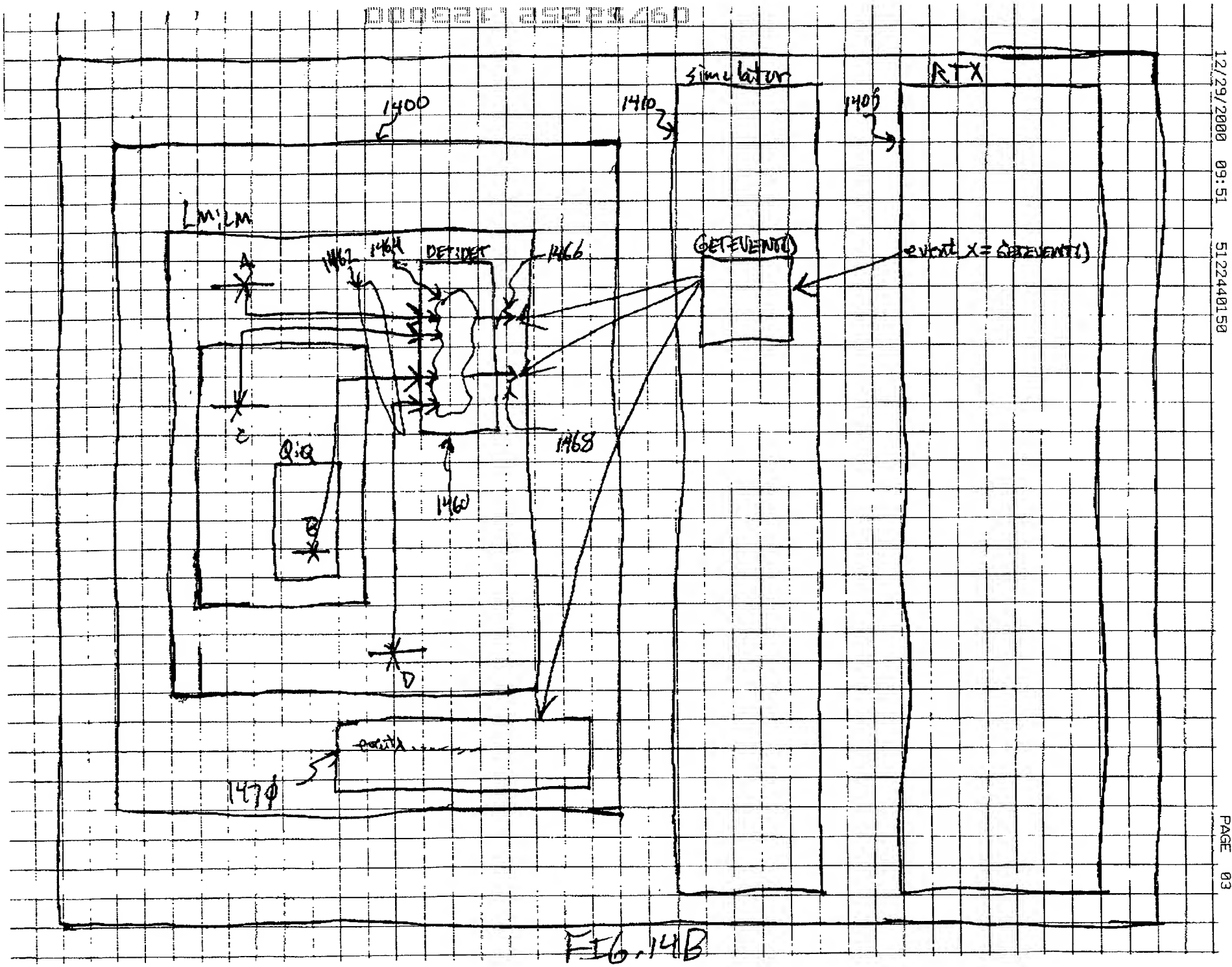

FIG. 14B

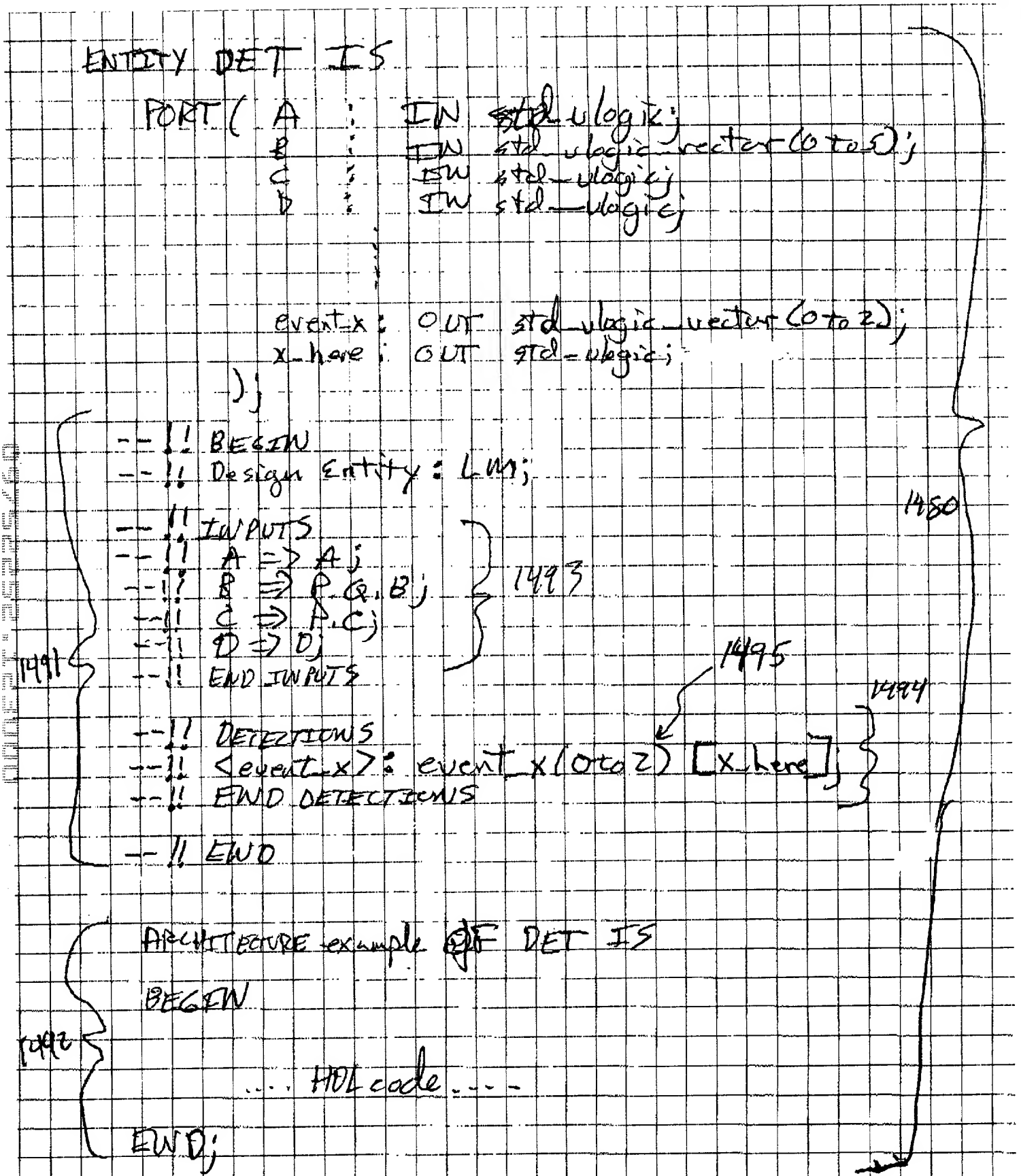
ENTITY DET IS
PORT(A : IN std_ulogic;
B : IN std_ulogic_vector(0 to 5);
C : IN std_ulogic;
D : IN std_ulogic;
event_x : OUT std_ulogic_vector(0 to 2);
x_here : OUT std_ulogic;
);
-- !! BEGIN
-- !! Design Entity : LUM;
-- !! INPUTS
-- !! A => A;
-- !! B => P.Q.B;
-- !! C => P.C;
-- !! D => D;
-- !! END INPUTS
-- !! DETECTIONS
-- !! <event_x> : event_x(0 to 2) [x_here];
-- !! END DETECTIONS
-- !! END
ARCHITECTURE example OF DET IS
BEGIN
... HDL code ...
END;
1491
1492
1493
1494
1495
1480

FIG. 14C